Real-Time Reconstruction and Rendering of Non-Polygonal Representations of Particle-Based Fluid Simulations

George Fyles
School of Design and Informatics
Abertay University
DUNDEE, DD1 1HG, UK

ABSTRACT

Context: Particle-based fluid simulations may increase realism in games, but present challenges such as the lack of a defined surface, which can be reconstructed using Signed Distance Fields (SDF).

Aim: To evaluate the effectiveness of novel surface reconstruction and rendering methods for SDFs of particle-based simulations.

Method: A test application will be developed in C++ with DirectX 12. A novel Nearest Neighbour Search combined with a newly proposed narrow band surface detection method utilizing a two-level bi-directional grid will expedite surface evaluation and construction of a texture storing voxel values and an accompanying acceleration structure for Ray Tracing (RT).

Results: Quantitative performance data and will be obtained and compared along with visual fidelity against that of a naïve implementation. It is anticipated that results will show a marked improvement in performance with minimal loss of visual fidelity.

Conclusion: This project will demonstrate whether some novel methods can be used effectively together, and show that 3D fluids are increasingly more feasible for modern games.

Keywords

Smoothed Particle Hydrodynamics, Signed Distance Field, Surface Reconstruction

1. INTRODUCTION

Convincing fluid has been a desirable feature in video games for decades, especially regarding realistic water. Since the 1990's, it has been standard practice for games to manipulate the vertex positions of a two-dimensional (2D) plane, creating the effect of lifelike water, informed by simulation data.

Real-time simulation is required for an immersive player experience as it allows for interactivity, however, simulating and rendering true three-dimensional (3D) fluids in real-time is still considered too computationally expensive, with some modern games opting to extract pre-baked animations from offline simulations to manipulate the plane (Liu, *et al.*, 2023). While visually impressive, the player cannot interact with the fluid. Another widely used approach is to limit real-time simulations to two dimensions, which while retaining reactivity, is limited in realism due to lack of realistic 3D details (Kellomäki, 2012), such as true to life splashes or submerged objects affecting turbulence. Clearly, research into simulating and rendering true 3D fluids in real-time is worthwhile.

Smoothed Particle Hydrodynamics (SPH) is a popular and widely researched method of simulating 3D fluids introduced by (Müller, Charypar and Gross, 2003). A particle-based solver, SPH works by discretizing the fluid's spatial differential operators and spatial field quantities. Navier-Stokes equations are then calculated for all particles, and collectively the particles describe the body of fluid. Figure 1 illustrates the concept of discretizing a fluid into particles.

Figure 1 – A volume of fluid is rendered as a continuum (Left) and as discretized particles (Right) (Müller, Charypar and Gross 2003).

Recently it has become possible to efficiently simulate millions of particles on the Graphics Processing Unit (GPU) owing to the inherently parallel nature of discretization (Liu, *et al.*, 2023). While there is more work to be done in this area, another challenge undergoing research is the reconstruction and rendering of the fluid surface. SPH offers no defined surface, so the surface must be reconstructed and rendered through other means, broadly divided into polygonal and non-polygonal methods, one being SDFs, discussed in Section 2.1.

1.1 Aim

This project aims to evaluate the effectiveness of a combination of novel methods for decreasing the latency of reconstructing and rendering surfaces of SDF representations of SPH simulations.

1.2 Objectives

- To review current techniques involved in reconstructing and rendering surfaces of SPH simulations
- To select some novel techniques for evaluation
- To develop an application utilizing the chosen techniques
- To gather performance data
- To examine and evaluate the data and visuals, compared to a baseline
- To make recommendations and suggest future work

2. BACKGROUND

2.1 Surface reconstruction

Methods for the rendering of surfaces of SPH data can predominantly be separated into polygonal and non-polygonal variants. In polygonal methods, the surface is rendered indirectly by first constructing a mesh around the volume, whereas non-polygonal methods render the volume directly.

Polygonal methods start by constructing some form of Scalar Field (SF), a mathematical function returning a single value for each location in space, regardless of reference frame. This function is used to determine an isosurface, which a mesh is then constructed around.

The most used method of mesh construction makes use of the Marching Cubes algorithm or its derivatives as in (Nishidate and Fujishiro, 2024). These methods divide the SF into a grid of cubes and define a set of possible topologies for the surface within a cube. The algorithm then 'marches' through the grid, determining how the isosurface intersects each cube and using a lookup table assigns one of the possible topologies to the cube, choosing the closest match. The result is a mesh estimating the isosurface topology, which can then be rendered with traditional rasterization techniques.

Non-polygonal methods render the fluid directly, without the intermediate mesh.

Screen space solutions such as point splatting or its derivatives may be used, as described by (Liu, *et al.*, 2023). This works by rasterizing the particles as point sprites and applying a blur, creating the appearance of a smooth surface.

A significantly more visually accurate procedure is to begin the same way as in polygonal methods by constructing a SF, but to omit the mesh construction stage and render the SF directly with RT techniques. Due to the historically large computational cost of RT compared to conventional rasterization strategies, polygonal methods have been favoured.

The work of (Liu, *et al.*, 2023) shows that while polygonal methods are still the fastest available, direct rendering of the SF through RT allows for improved accuracy and near photo-realism, and due to advances in hardware-accelerated RT available on modern hardware, is becoming more feasible for games. A demonstration of this ran at 25 frames per second with 2.5 million particles on a single RTX 3090 GPU.

Therefore, further research into real-time reconstruction and rendering of non-polygonal representations of SPH simulations is valuable.

Polygonal and non-polygonal methods may require a SF to be constructed, and different types of SF have been used. (Müller, Charypar and Gross, 2003) suggest using a Smoothed Colour Field (SCF). A *colour field* is where field values are 1 at particle positions and 0 elsewhere. This is smoothed using a kernel function.

A different approach introduced by (Zhu and Bridson, 2005) is to use an SDF. In this scheme, field values are determined by a function returning the signed distance to the implicit isosurface; a point on the isosurface has a value of 0, a point outside has a positive value, and a point inside has a negative value. While the SCF method produces unappealing bumps, the SDF method gives a much smoother and accurate representation. SDFs also lend themselves well to RT and are used by (Liu, *et al.*, 2023).

2.2 Nearest Neighbour Search

Any given field value within an SDF representation of a particle-based simulation is determined by the weighted average of surrounding particle positions and their neighbours within a fixed support radius. A naïve approach to finding neighbouring particles would involve checking every particle against every other particle, but this is bad practice for two reasons: firstly, two particles far apart should not affect one another, and secondly this brute force method has time complexity $O(N^2)$. A fast method of Nearest Neighbour Search (NNS) is therefore required.

(Hoetzlein, 2014) improved upon earlier work, introducing a fast fixed-radius NNS strategy. The volume is partitioned into a grid with cell side lengths equal to the support radius. To find neighbours within this radius, only particles in the adjacent $3^3=27$ cells need be queried, shown in Figure 2. A counting sort using a prefix scan sorts particle indices according to their cells for fast grid search.

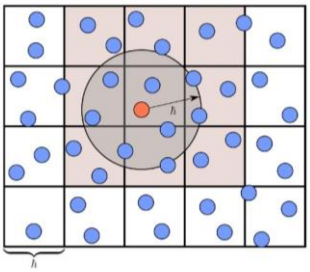

Figure 2 – Only particles in the adjacent cells need be checked. (Liu, *et al.*, 2023).

(Liu, *et al.*, 2023) make further improvements on this method, using an updated single-pass global prefix scan, and instead of sorting a list of particle indices, the particle data itself is moved in memory, ensuring the data is spatially coherent for an improved cache hit rate when accessing neighbouring particles.

The same paper also proposes a novel method for efficiently storing a neighbour list per particle representing each neighbour with a single bit, and a new algorithm for fast traversal of this list using bitmask operations.

2.3 Narrow Band Methods

In both polygonal and non-polygonal methods, surface reconstruction can be significantly sped up if evaluation is limited only to those particles representing the surface of the fluid, the *narrow band* region. Therefore, in recent years researchers have investigated fast and precise methods of surface particle identification.

The method suggested by (Müller, Charypar and Gross, 2003) identifies a surface particle when the SCF value is above a certain threshold. This requires the threshold to be fine-tuned, and even when this is done non-surface particles may be identified as surface particles and vice versa.

An accurate and parallelisable method was introduced by (Yang and Gao, 2020), which is surprisingly simple: the cells of the spatial grid used for NNS track how many particles they contain, and a cell is found to be at the surface if it is not empty and at least one adjacent cell is empty. Particles within such a cell are designated as surface particles.

While this offers performance gains, (Nishidate and Fujishiro, 2024) note that detecting surface cells remains a time-consuming process for high resolution grids. They expedite the procedure using a two-level bi-directional grid consisting of a fine grid, said to contain *cells*, and a coarse grid containing *blocks*. While the fine grid works the same as in previous methods, blocks in the coarse grid track non-empty cells within them. If a block is completely empty or completely full, it is not a surface block. This allows surface cell detection to be accelerated as only cells in surface blocks need be checked, as shown in Figure 3.

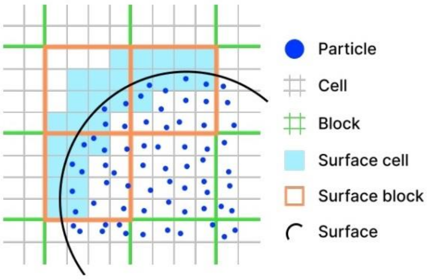

Figure 3 – Bi-directional grid for accelerated surface cell detection (Nishidate and Fujishiro, 2024).

2.4 Discretization and rendering of SDFs

When ray tracing an SDF representation of geometry, a naïve implementation would evaluate the isosurface for every ray. A more effective solution is to discretize the SDF into a grid of voxels and store this as a 3D texture to be sampled in the RT step.

(Söderlund, Evans and Akenine-Möller, 2022) conducted an analysis of several methods for storing and traversing the grid, finding that a Sparse Voxel Set (SVS) performed best. In this scheme an Axis-Aligned Bounding Box (AABB) is created for each voxel from which a bounding volume hierarchy is built for hardware-accelerated RT. When a ray intersects an AABB an isosurface intersection test such as sphere tracing is performed and normals can be found through central differencing or similar methods. A similar method, Sparse Brick Set (SBS) instead creates an AABB for *bricks* of 7x7x7 voxels, allowing efficient use of High-level Shader Language functions for sampling SDF values. SBS is slower but uses less memory for storing the AABBs.

The same paper also presents a novel analytic voxel intersection test, however from their results it appears sphere tracing is also sufficient for real-time in simple scenes.

3. METHOD

This project will build an application to evaluate the effectiveness of a combination of a few of the novel techniques as not yet seen in the literature, covering the two general stages of SDF surface reconstruction and SDF discretization and rendering. The application will be developed in C++ using DirectX 12 to leverage DirectX Raytracing hardware-accelerated RT as in (Söderlund, Evans and Akenine-Möller, 2022). The majority of the procedures tested will run in compute shaders on the GPU.

3.1 Particle data

Since the focus of the project will be on the reconstruction and rendering of SPH simulation data rather than the simulation itself, during early development all that is needed is a randomly generated list of positions each frame. Further into development either a pre-existing library for SPH simulation on the GPU – there are several suitable ones – will be integrated, or an in-house solution will be implemented following a similar method to that presented by (Liu, *et al.*, 2023).

3.2 SDF surface reconstruction

For fast surface reconstruction, a method similar to that proposed by (Liu, *et al.*, 2023) will be utilized to efficiently sort particles in memory according to a spatial grid and build neighbour lists, however a two-level bi-directional grid approach as introduced by (Nishidate and Fujishiro, 2024) will be used. As presented in that paper, this will be used to rapidly detect surface cells. Voxels will then be sampled from surface cells only, which should yield an improvement over the method of (Liu, *et al.*, 2023), which samples voxels in all non-empty cells. During SDF evaluation the novel bit traversal algorithm as per (Liu, *et al.*, 2023) will likely be used for fast NNS. That method does have a limitation in that it assumes a cell can only contain up to 32 particles, and if improvements to address this issue can be found they will be applied.

3.3 SDF discretization and rendering

The project will aim to evaluate use of the SVS and SBS schemes for SDF sampling and traversal put forward by (Söderlund, Evans and Akenine-Möller, 2022), using sphere tracing for intersection testing. AABBs will be created for each voxel or for each brick within surface cells as determined using the two-level bi-directional grid as mentioned above.

Development will focus first on ensuring the core systems work through rendering the fluid as opaque geometry, though if time allows it, more advanced RT techniques for photorealism will be implemented as put forward by (Liu, *et al.*, 2023).

3.4 Testing

Testing will be conducted using an RTX 4070 Ti GPU representing the latest generation of consumer hardware, as none of the studies mentioned have used this generation in their testing.

Quantitative performance data will be gathered using a graphics debugger, most likely NVIDIA Nsight. Performance metrics gathered will include frames per second, durations of individual procedures, memory usage and cache hits, hardware unit throughputs, GPU warp occupancy, and anything else deemed necessary.

All stages of the pipeline shall be complemented by a naïve implementation, for which the same metrics will be recorded as a baseline for direct comparison.

The application will be tested under different conditions by varying parameters such as particle numbers, grid resolutions and voxel/brick sizes. Various test scenes will be developed, with an orbiting camera to eliminate any view dependent bias, to determine how particle behaviour affects performance.

As the surface will be sampled from a 3D texture, visual fidelity will also need examination, by comparing renders of a predefined set of particles using voxels of varying sizes against a non-discretized naïve render. Surface normals and visual artifacts are examples of things to consider.

4. SUMMARY

The efficient rendering of 3D fluid simulations in games is becoming an ever more popular avenue of research as developers strive to increase realism and player immersion. This project will evaluate the effectiveness of a novel NNS method combined with a novel two-level bi-directional grid narrow band method, a combination not yet investigated, for the reconstruction of the isosurface of SDF representations of SPH simulations. The project will also evaluate the effectiveness of leveraging the new two-level grid for the construction of SVS and SBS for RT, a use not suggested by the authors.

It is hoped the data gathered in this study will prove useful to other researchers investigating this topic.

5. REFERENCES

Hoetzlein, R. (2014) 'Fast Fixed-Radius Nearest Neighbors: Interactive Million-Particle Fluids', *The GPU Technology Conference 2014*, San Jose (United States) 24-27 March. Available at: https://ramakarl.com/pdfs/2014_Hoetzlein_FastFixedRadius_Neighbors.pdf (Accessed 15 October 2024).

Kellomäki, T. (2012) 'Water simulation methods for games: a comparison', *MindTrek '12: Proceeding of the 16th International Academic MindTrek Conference*, Tampere (Finland), 3-5 October. New York (United States): Association for Computing Machinery, pp. 10-14. Available at: https://doi.org/10.1145/2393132.2393135

Liu, B. *et al*. (2023) 'Building a Real-Time System on GPUs for Simulation and Rendering of Realistic 3D Liquid in Video Games', *SIGGRAPH '23: ACM SIGGRAPH 2023 Courses*, Los Angeles (United States), 6-10 August. New York (United States): Association for Computing Machinery. Available at: https://doi.org/10.1145/3587423.3595537

Müller, M., Charypar, D. and Gross, M. (2003) 'Particle-based fluid simulation for interactive applications', *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. San Diego (United States), 26-27 July. Goslar (Germany): Eurographics Association, pp. 154-159. Available at: https://matthias-research.github.io/pages/publications/sca03.pdf (Accessed 15 October 2024).

Nishidate, Y. and Fujishiro, I. (2024) 'Efficient Particle-Based Fluid Surface Reconstruction Using Mesh Shaders and Bidirectional Two-Level Grids', *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1), pp. 1-14. Available at: https://doi.org/10.1145/3651285

Söderlund, H., Evans A. and Akenine-Möller, T. (2022) 'Ray Tracing of Signed Distance Function Grids', *Journal of Computer Graphics Techniques*, 11(3), pp. 94-113. Available at: https://www.jcgt.org/published/0011/03/06/ (Accessed 15 October 2024).

Yang, W. and Gao, C. (2020) 'A completely parallel surface reconstruction method for particle-based fluids', *The Visual Computer*, 36, pp. 2313–2325. Available at: https://doi.org/10.1007/s00371-020-01898-2

Zhu, Y. and Bridson, R. (2005) 'Animating sand as a fluid', *ACM Transactions on Graphics*, 24(3), pp. 965-972. Available at: https://doi.org/10.1145/1073204.1073298