Received December 1, 2017; Accepted March 11, 2017

International Journal of Asia Digital Art&Design

Local Density Calibration for Position Based Fluids

Taihei Meno
Kyushu University
t.meno1027@kyudai.jp

Reiji Tsuruno
Kyushu University
tsuruno@design.kyushu-u.ac.jp

Abstract

This paper propose a improved PBF(position based fluid model)-based method for simulating realistic incompressible fluid flow. Our goal is to present simulation method of fluid phenomena as realistic as possible in realtime to many applications that require a high frame rate. Rapid fluid simulations are important for realizing real-time applications in computer graphics. Position based fluids (PBF) has been suggested as a practical method for real-time simulations of incompressible fluids. This method is based on the SPH(smoothed particle hydrodynamics) and realizes incompressibility by restricting the density. Furthermore, it reduces the calculation costs by allowing large time steps. However, because it iteratively calculates the density constraints, PBF is impractical for applications requiring high resolution and high frame rate. To increase the simulation speed of PBF, we replace the iterative density calculation by a density model. In the conventional method, iterating few density calculations compresses the simulated fluid. To resolve this problem, we consider the particle density and the direction of the fluid flow. We develop a computational model based on this hypothesis and verify its usefulness in simulations. The method suppresses the fluid compression even in a non-iterative model. We conclude that the proposed computational model effectively maintains a constant density in PBF.

Keywords: fluid simulation, real time simulation, position based fluids, density correction model

1 Introduction

This paper propose a improved PBF(position based fluid model)-based method for simulating realistic incompressible fluid flow. Our goal is to present simulation method of fluid phenomena as realistic as possible in realtime to many applications that require a high frame rate. In this way, we focus on the simulation method to improve the quality of real-time applications by drawing realistic fluids at high speed with computer graphics. Recent virtual reality technology demands high-frame-rate video and color graphics for fast fluid simulations.

Fluids can be most realistically expressed in physics simulations such as smoothed particle hydrodynamics (SPH) [1] and moving particle semi-implicit (MPS) simulations [2]. However, although these techniques capture the true behaviors of fluids, they are impractical in real-time applications because preserving the incompressibility of the fluid is calculation-intensive. Therefore, in order to represent realistic fluids in real-time applications, researches on speeding up computation of fluid simulation are being conducted. J.Stam proposed a high-speed fluid simulation method using a grid method for games [3][4]. And Muller showed that SPH can be used for interactive fluid simulation by using a low stiffness equation of state[5]. However, these methods are not visually realistic, and there is a problem that calculation time takes too much for practical use. Ando focuses on fluid surface and presents the method of make fluid sheet from adaptively sampled anisotropic particle[6][7][8]. Yoshino controls the liquid particles behavior[9] using shape-matching[10] and key frame target driven method[11].

A solution is offered by position based fluids (PBF) [12], which realistically simulates fluids at high speed. To maintain an incompressible fluid, this method constrains the particle density to a constant. A particle is moved by correcting its position to satisfy the constraint condition. By geometrically determining the particle position, PBF performs stable calculations with a large calculation step (which confers the speed advantage). However, PBF requires iterative calculations to satisfy the constraint condition of the particles. When the number of iterative calculations is small, the incompressibility of the fluid is not maintained and the calculation result deviates from the true solution. In this research, we develop a computational model that replaces the density constraints of PBF while delivering realistic, high-speed simulations of fluid flows.

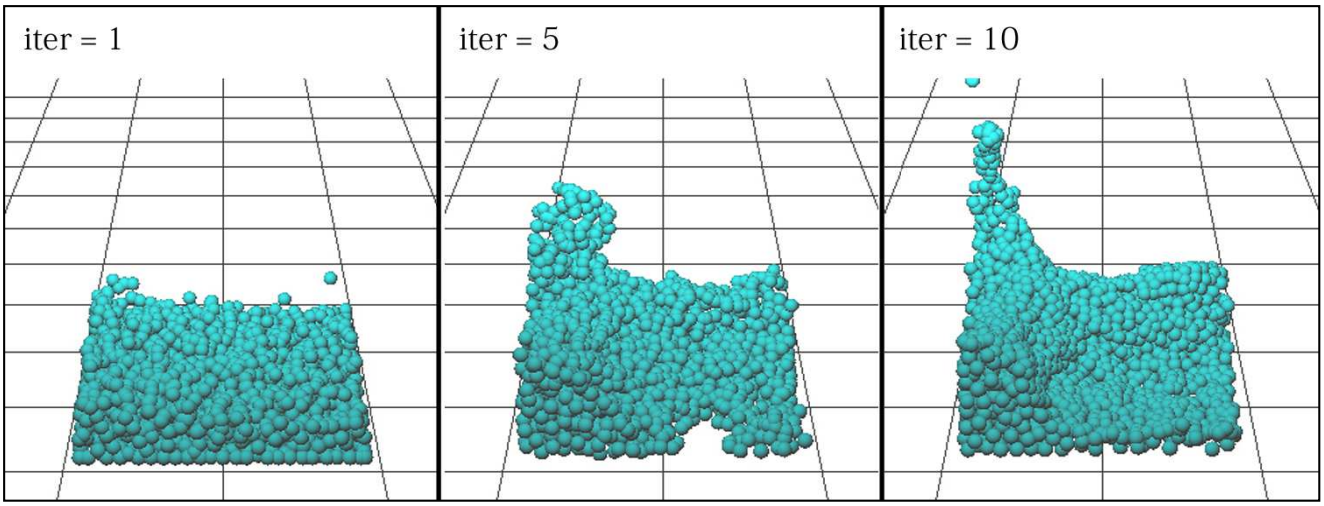

Figure 1 Simulation results after different numbers of iterations

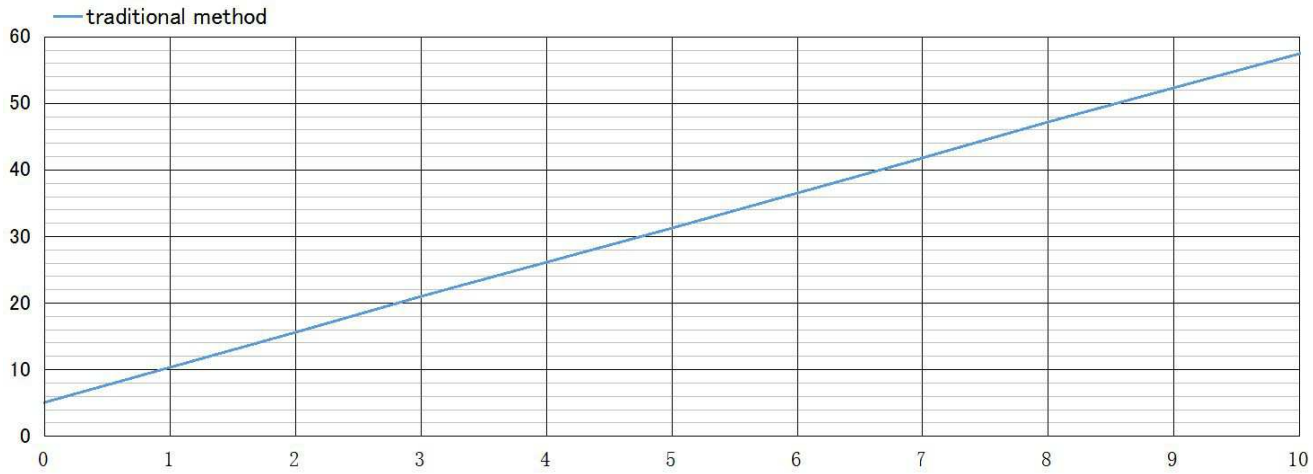

Figure 2 Simulation time per frame

2 Local Density Correction Model

Figure 1 compares the simulation results at different iteration numbers of the same time step. Figure 2 plots the relationship between iteration number and simulation time per frame. As shown in Figure 1, the fluid motion becomes more realistic as the number of repetitions increases. When the number of iterations is small, the fluid is compressed and the flow is gentle. However, as the number of iterations increases, the calculation time increases proportionally (see Figure 2). In PBF, a small number of iterations results in violations of the density constraint, and the fluid loses its incompressible property. As reducing the iteration number is crucial for reducing the computational costs, a method that maintains incompressibility under fewer iterations is required.

To accelerate the simulation calculation, we propose an alternative computational model that avoids the iterative density constraint calculations. This method considers the density of each particle. The sum of the forces received from particles within the effective radius of a certain particle push the particle in the direction of lower density, ensuring the constant density condition. This calculation is given by Eq. 1. The forces reaching an arbitrary particle within its effective radius increase with decreasing distance between the particles. Moreover, the greater the density differences among the particles within the effective radius, the greater is the force. If the particles within the effective radius are less dense than the arbitrary particle, the force is directed toward the arbitrary particle. In contrast, higher density of the surrounding particles directs the force away from the arbitrary particle (Eq. 2).

$$f_i^{density} = -K\sum_j \rho h\left(1-\frac{\left|x_i - x_j\right|}{h}\right), \quad (1)$$

where h and x are the effective range and position vector of the particle, respectively. The pressure difference ρ between two particles *i* and *j* is computed as

$$\rho = \frac{\rho_j - \rho_i}{\rho_0}, \qquad (2)$$

where ρ_0 is the initial density and K is an arbitrary constant.

3 Algorithm

The algorithm of our proposed method is shown in Figure 3. In this simulation, v and x denote the speed and position vectors of particle i, respectively, f^{ext} is the external force, and $f^{density}$ is our density correction model. Δt denotes the time step, λ is the Lagrangian multiplier, and Δp is the position correction.

Our method runs the calculation model before correcting the position to satisfy the density constraint. This order is imposed because moving the particle in the direction that smooths the density obtains a provisional position that partially completes the position correction. We first calculate $f^{density}$ by our density-complement model. Next, we calculate the external forces and find the temporary velocities and positions under the external forces and our density correction model. The position correction Δp is then calculated from the temporary position, and the particle position and velocity are updated in readiness for the next time step.

```
For all particles i do
        Calculate f^density
End for

For all particles i do
        apply forces v_i = v_i + (f^ext + f^density)Δt
        predict position x_i^* = x_i + v_iΔt
End for

For all particles i do
        find neighboring particles
End for

While iter < solverIterations do
        For all particles i do
                calculate λ_i
        End for

        For all particles i do
                calculate Δp_i
        End for

        For all particles i do
                Update position x_i^* = x_i^* + Δp_i
        End for
End while

For all particles i do
        Update velocity v_i = (x_i^* - x_i) / Δt
        Update position x_i = x_i^*
End for
```

Figure 3 Outline of Algorithm 1 loop

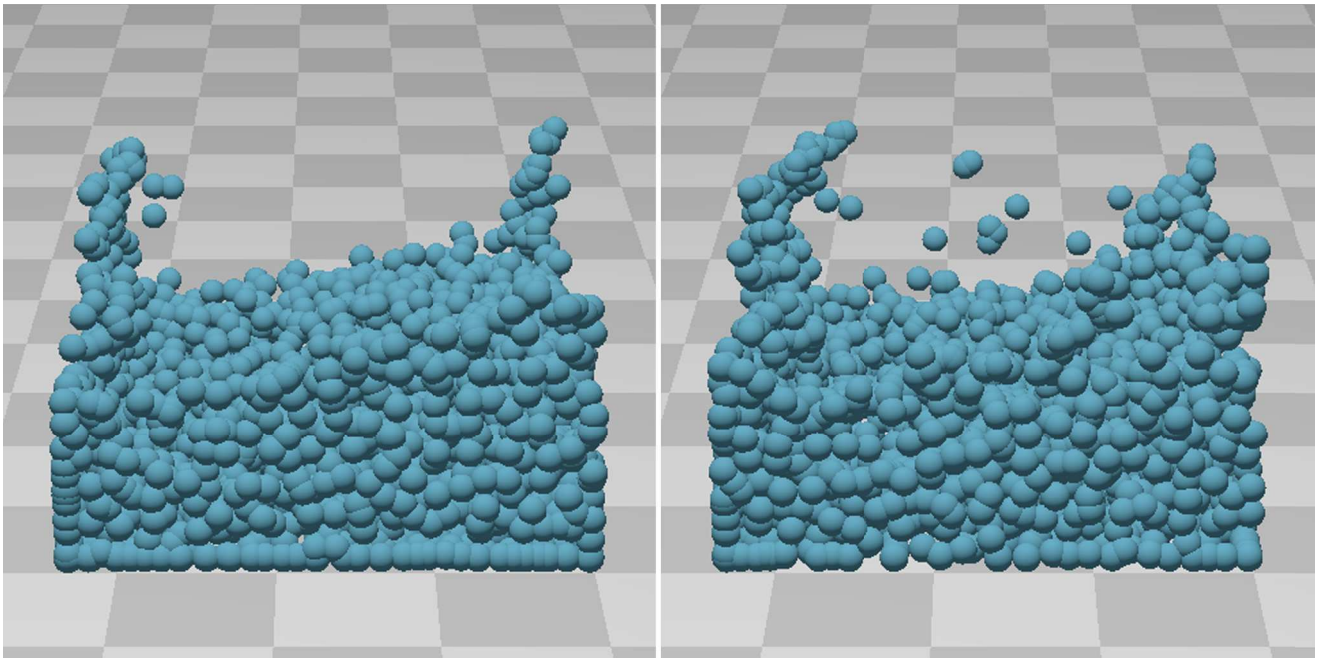

Figure 4 Comparisons of simulation results (1 iter, Left: conventional method, Right: our method)

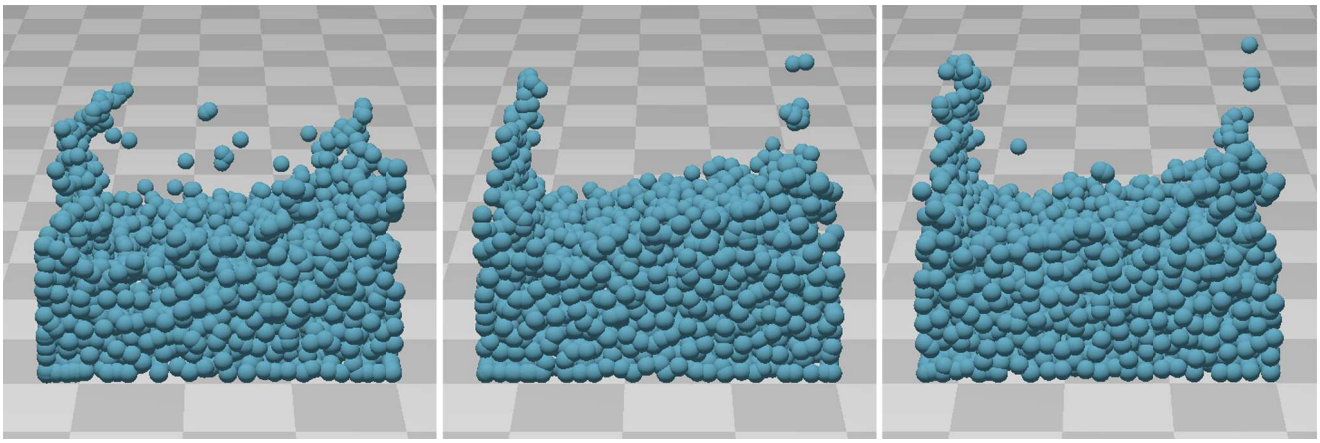

Figure 5 Effectiveness of our method
(Left: our method -1 iter, Center: conventional method –2 iters, Right: conventional method –3 iters)

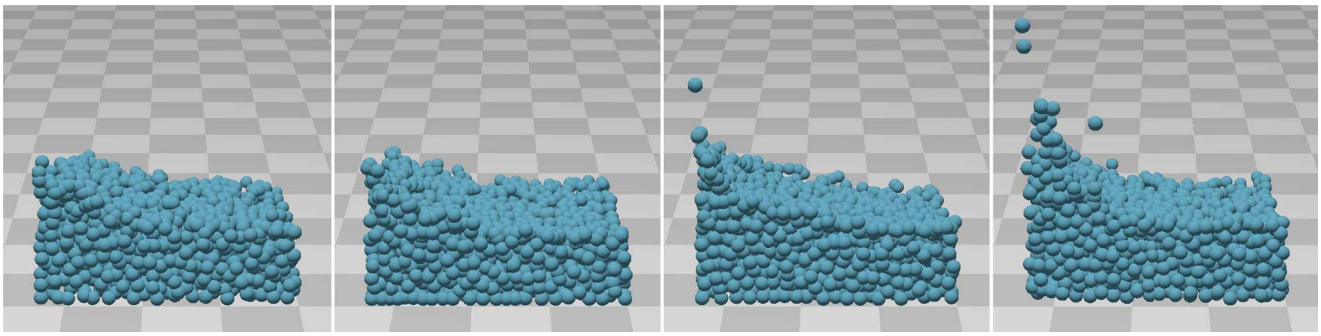

Figure 6 Flow attenuation
(From left: our method -1 iter, traditional method –1 iter, conventional method –2 iters, conventional method –3 iters)

4 Result

Figure 4 presents the simulation results of the conventional and proposed methods at the same time under the same conditions. In the conventional method, the few repetitions of the density constraint lead to fluid compression, especially at the walls. Our method improves the compression problem at these same sites. In other words, unlike the conventional method, our method maintains incompressibility even when the number of iterations is small. Our method preserves incompressibility by applying a force in the direction that minimizes the density differences among the particles. Also, because our method is unaffected by iteration number, it yields satisfactory results after just a few iterations.

Figure 5 compares the simulation results of our method and the conventional method after two and three iterations. As the number of iterations increases, the compression problems in the conventional method are gradually improved. Comparing the result of the proposed method with one iteration and the conventional method with two iterations, the compression problem is solved by the proposed method as well.

Figure 6 shows how the flow momentum changes with number of iterations. In the conventional method, more iterations not only improve the compression problem of the fluid, but also yield more realistic fluid flows. Our proposed method less effectively improves the flow momentum, because the density complement model corrects the particle position by density considerations alone, and ignores the direction of the fluid flow.

Figure 7 plots the relationship between the iteration number and simulation time in the conventional and proposed methods. Our method increases the simulation time per iteration by approximately one-sixth that of the conventional method. Overall, however, the proposed method can decrease the calculation time by reducing the number of iterations required for suppressing the particle compression in the conventional method. We conclude that our method effectively increases the calculation speed of fluid simulations.

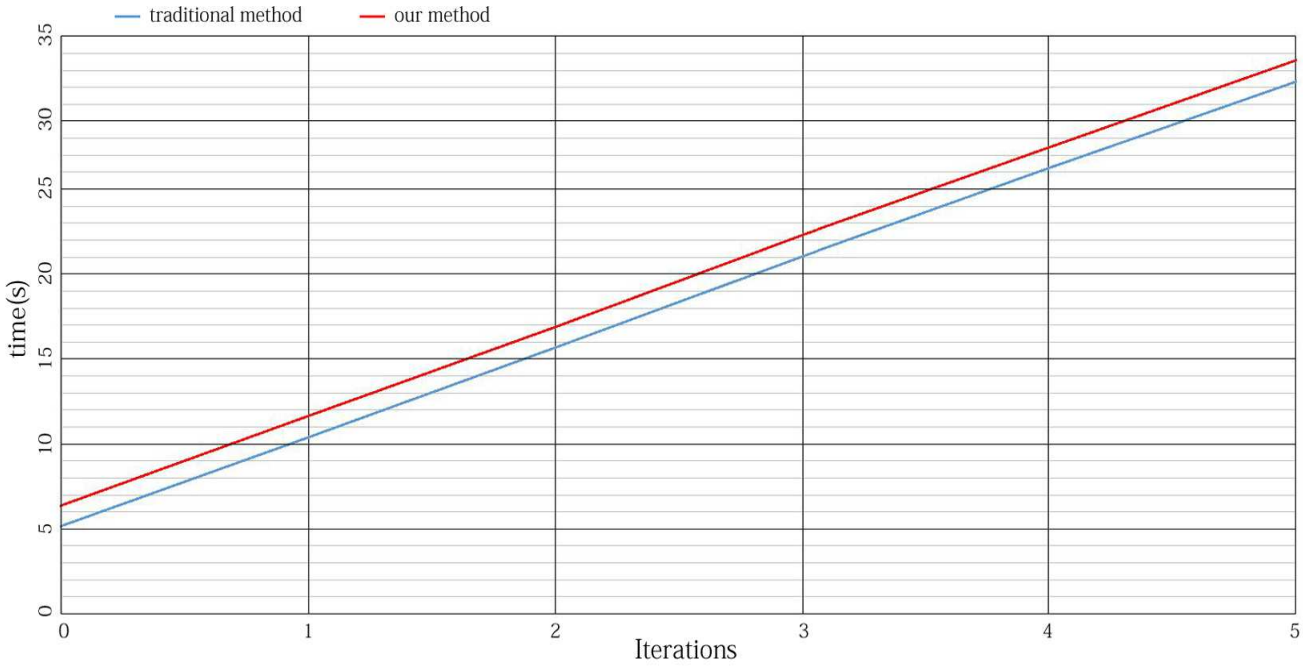

Figure 7 Simulation time per frame

5 Limitations and Future Work

This research confirmed that our proposed density-complement model effectively solves the compression problem in the conventional method. Our model also reduces the calculation time by reducing the number of iterations required for a realistic result. .However, our method little improved the fluid flow over the conventional method, because it does not consider the direction of the fluid flow. To solve this problem, the flow direction must be incorporated into our density complement model. This task is reserved for future work. The new calculation model must also sufficiently consider the increased time and the validity of the simulation results.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 26350012.

References

[1] Monaghan.J.J, *Smoothed particle hydrodynamics*, Annual Review of Astronomy and Astrophysics, vol. 30, No. 1, pp. 534-574, 1992

[2] Seiichi Koshizuka, Yoshiaki Oka, *Moving-Particle Semi-Implicit Method for Fragmentation of Incompressi-ble Fluid,* Nuclear Science and Engineering, Vol. 123, No. 3, pp. 421-434, 1996

[3] Jos Stam, *Stable fluids*, SIGGRAPH '99 Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 121-128, 1999

[4] Jos Stam, *Real-time fluid dynamics for games*, Proceedings of the game developer conference, vol 18, pp. 25, 2003

[5] Matthias Müller, David Charypar, Markus Gross, *Particle-Based Fluid Simulation for Interactive Applications*, SCA'03 Proceedings of the 2003 ACM SIGGRAPH/ Eurographics symposium on Computer animation, pp.154-159, 2003

[6] Ryoichi Ando, Reiji Tsuruno, *A particle-based method for preserving fluid sheets,* SCA '11 Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer animation, pp. 7-16 (cit. on p. 30), 2011

[7] Ryoichi Ando, Nils Thurey, Reiji Tsuruno, *Preserving Fluid Sheets with Adaptively Sampled Anisotropic Particles,* IEEE Transaction on Visualization and Computer Graphics, vol.18, no.8, pp.1202-1214, 2012

[8] Ryoichi Ando, Nils Turey, Cris Wojtan, *Highly Adaptive Liquid Simulations on Tetrahedral meshes*, ACM Transactions on Graphics, vol.32, no.4, article103, 2013

[9] Jun Yoshino, Reiji Tsuruno, *Interactive fluid control by shape matching*, The Journal of the Society for Art and Science, Vol. 13, No. 3, pp. 169-176, 2014

[10] Matthias Muller, Bruno Heidelberger, Matthias Teschner, *Meshless Deformations Based on Shape Matching*, ACM Transactions on Graphics, vol.24, no.3, pp.471-478, 2005

[11] Adrien Treuille, Antoine McNamara, Zoran Popovic, Jos Stam, *Keyframe control of smoke simulations,* ACM Transaction on Graphics, vol.22, 3, pp.716-723, 2003

[12] Miles Macklin, Matthias Müller, *Position Based Fluids*, ACM Transactions on Graphics, Vol. 32, Issue 4, pp.104:1-104:12, 2013.