

# Particle-Based Fluid Analysis for Position-Based Flow Data

Lieyu Shi

*School of Natural Sciences and Mathematics  
University of Houston  
Houston, Texas 77004  
Email: shilieyu91@gmail.com*

Guoning Chen

*School of Natural Sciences and Mathematics  
University of Houston  
Houston, Texas 77004  
Email: guoning.chen@gmail.com*

**Abstract**—Particle-based fluid simulation (PFS), such as Smoothed Particle Hydrodynamics (SPH) and Position-based Fluid (PBF), is a mesh-free method that has been widely used in various fields, including astrophysics, mechanical engineering, and biomedical engineering for the study of liquid behaviors under different circumstances. Due to its meshless nature, most analysis techniques that are developed for mesh-based data need to be adapted for the analysis of PFS data. In this work, we study a number of flow analysis techniques and their extension for PFS data analysis, including the FTLE approach, Jacobian analysis, and an attribute accumulation framework. In particular, we apply these analysis techniques to free surface fluids. We demonstrate that these analyses can reveal some interesting underlying flow patterns that would be hard to see otherwise via a number of PFS simulated flows with different parameters and boundary settings. In addition, we point out that an in-situ analysis framework that performs these analyses can potentially be used to guide the adaptive PFS to allocate the computation and storage power to the regions of interest during the simulation.

## 1. Introduction

Fluids (liquids, gases, plasmas and plastic solids) are ubiquitous in our lives. The development of various fluid models and their simulations via the research in the field of computational fluid dynamics (CFD) have been widely applied in various scientific, engineering and entertainment applications to help domain experts and practitioners study fluids with different characteristics for the needs of their specific applications. Automotive and aircraft design, mechanical engineering, environmental science, oceanography, climate study, plasma physics, movies, commercials and games are just a few places where fluid simulations have been applied.

To identify dynamics (or features) that are of relevance to the specific applications is of paramount importance for the aforementioned applications. For instance, vortical flow is of interest to the study of mixing of gases [8] and the diagnosis of certain cardiac diseases [23]. Most of these features can not be revealed via direct rendering of the simulation results. The analysis and visualization of the

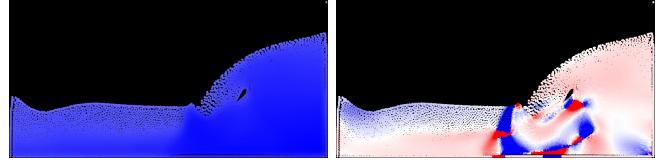


Figure 1. The direct rendering of particles of a 2D SPH simulation (left) and the visualization with different colors assigned to particles based on their characteristics (right).

simulated fluids can help impose visual clues to the places with those important dynamics, largely simplifying the task of interpreting the simulations results. See Figure 1 for an example. Consequently, analysis and visualization of simulated fluids has been a popular research area in the visualization community in the past few decades. However, most existing techniques [15], [24] are for fluids stemming from Eulerian simulation framework. That is, quantities are defined on fixed mesh grid points or at the centers of cells, so that well-established interpolation techniques (e.g., linear interpolation) can be employed to obtain quantities elsewhere in the domain. In contrast, fluids resulted from the particle-based simulation (or Lagrangian framework), denoted by PFS, do not have explicit and fixed neighborhood information. That said, analysis techniques for mesh-based fluid data cannot be directly applied to the analysis of PFS data. In addition, there is little attention on the free surface fluid simulations that have been applied in various computer graphics applications. Understanding the characteristics of the free surface fluids, especially their underlying behaviors may help develop efficient fluid simulators that assign computation and memory resources to places where important, critical features occur [31], [44]. However, free surface fluids typically possess complex surface shapes changing over time, making their analysis challenging. These challenges motivate this work.

In this work, we focus on the particle-based fluid simulation (PFS) data and its analysis and visualization. Specifically, we adapt a number of well-known analysis and visualization techniques, including FTLE [21] and asymmetric tensor field analysis [46], for the analysis of PFS data that depicts free surface fluids. In addition, we apply a recently

introduced accumulation framework to study the overall (or average) behaviors of the individual particles as well as their relative behaviors with respect to their neighbors. We implemented the visualization and analysis framework in OpenGL coupled with GLUI library, and able to run the on-line simulation from generated file data.

## 2. Related Work

Smoothed Particle Hydrodynamics (**SPH**) was first proposed by Gingold and Monaghan [18] and Lucy [25], and then experienced extensive improvements and applications. Dependent on viscosity or not, SPH can achieve good simulation results in highly viscous flow (like jet buckling [13], sand [51] and [10]) and inviscid flow [12]. Based on simulation scenarios, SPH can also be used to simulate gas [34], smoke [32] and bubbles [37]. There are tremendous applications and progress involved with SPH. We refer interested readers to a sequence of survey papers [27], [28], [43], [47] for a complete overview of this field.

Enforcing incompressibility is always numerically challenging for SPH, because achieving divergence-free velocity field is computationally expensive and numerically unstable, especially with boundary conditions. Cummins et al. [38] introduced a pressure projection method to compute an intermediate velocity field and then projected it onto a divergence-free space by solving a pressure Poisson equation. However, it is time-consuming to solve Poisson equation with conjugate gradient solver and it doesn't scale well for large system. Becker et al. [4] introduced a weakly compressible SPH (WCSPH) for free surface flows. Due to the exponential term for pressure calculation, the time step size is greatly limited, which hampers its application in practice. A prediction-correction incompressible SPH (PCISPH) was proposed by Solenthaler et al. [39], which achieves an order of magnitude speed up while keeping the comparable results to WCSPH. Besides the divergence-free velocity constraint, incompressibility can also be partially achieved by the density constraint [3], [7] or volume-preserving [41]. However, these methods still suffer from the limited time steps to some extent.

Position-based dynamics is a prospering method to enable large time step by focusing on only the positions of particles solved by constraints other than accelerations and velocities. It was first introduced to the graphics community by Muller et al. [30] to simulate rigid bodies and cloths, and later improved and extended to many other simulation problems, such as rigid-body coupling [14] for sake of realistic effect and computational convenience. We refer interested readers to the survey papers by Bender et al. [5], [6] to acquire more details. Macklin et al. [26] used position-based method to achieve incompressibility by solving a density constraint, which enables both large time step and incompressibility effect for fluid simulation. However, it is still difficult to combine collision detection and incompressibility, especially for sharp boundaries.

Surface representation is also important for particle-based simulation. There exist many methods, like the color-

based surface tension used in WCSPH [4] and the ray-tracing method applied by PCISPH [39]. We also refer readers to Fang et al. [16] which provides a rigid numerical method to distinguish surface particles. In PBF [26], particle anisotropy is firstly computed using anisotropic kernels [45], and then the fluid surface is constructed by the screen-space filtering technique [42].

Little work has been done on particle-based data analysis, and most of the work is concerning visualization for large and time-varying particle datasets [11], [19], [40]. J. Chandler et al. [9] introduced an interpolation-based pathline tracing, but the computation is expensive and can't tell physically intuitive information for particle separation. In the meantime, the Finite Time Lyapunov Exponent(FTLE) that measures the rate of separation of flow particles during their transportation has gained increasing attention in the flow visualization community [17], [20], [21], [36]. Considering the Lagrangian nature of the PFS data and the FTLE computation framework, it is natural to apply FTLE computation for the analysis of PFS data. Agranovsky et al. [1] proposed to estimate the FTLE structure using sparse particles with the moving least squares (MLS), but his work only focuses on 2D while in 3D both kernel searching and least square fitting for each particle is more complicated. Our work is able to apply FTLE in 3D simulation data, and with an isotropically dynamic radius, our method is robust enough to get enough and minimal neighbor particles for FTLE computations. Besides, Agranovsky et al. [2] later proposes a pathline-based representation for Lagrangian fluid flow data and reconstructs pathline surface, which is totally different from what we did here.

## 3. Methodology and Implementation

### 3.1. Simulation Techniques

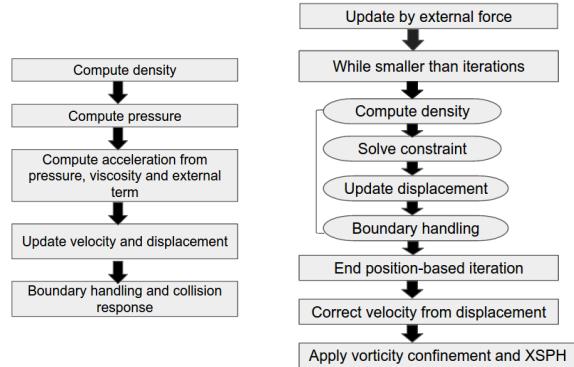


Figure 2. Pipeline comparisons between SPH (left) and position-based fluids (PBF) (right).

The conventional SPH represents the Navier-Stokes equation into discretized Lagrangian particles with certain finite-support kernel functions, and each particle conveys information of mass, density, velocity and vorticity. As in

the basic pipeline of SPH of Figure 2 (left), enforcing compressibility highly depends on pressure computation, which is either achieved by weakly compressible SPH [4], [29] using

$$\begin{aligned} p &= k\rho \\ p &= k(\rho - \rho_0) \\ p &= B\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) \end{aligned} \quad (1)$$

or pressure projection for a divergence-free velocity field [38].

Position-based method (PBF) was firstly introduced into computer graphics community in [30] and then was applied to achieve incompressible fluid by enforcing the constant-density constraint in [26]. A prediction of particle displacement is iteratively updated by solving the non-linear density constraint function

$$C_i(\mathbf{p}_1, \dots, \mathbf{p}_n) = \frac{\rho_i}{\rho_0} - 1 \quad (2)$$

The particle cohesiveness and numerical dissipation is improved by artificial pressure and vorticity confinement. The general pipeline is illustrated by the right flow chart in Figure 2.

In our practical experiment, we did 3D flows by using PBF framework. Since boundary handling in [26] that uses signed distance causes serious particle clumping, we adopted the boundary particle handling from [3] and relaxed density invariant method from [22] to improve the robustness in collision detection and simulation instability.

### 3.2. Analysis Framework

We introduced basically two analysis techniques for PFS data, FTLE and integral attribute.

**3.2.1. FTLE Analysis.** The finite-time Lyapunov exponent (or simply FTLE) measures the rate of flow separation during transportation. The FTLE computation results in a scalar field whose ridges correspond to the transportation barrier where the flow flux is negligible. Given a flow  $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}, t)$ , The FTLE value at each sample point  $\mathbf{x}$  can be derived from the flow map deformation  $\mathbf{D} = \nabla \varphi_{t_0}^{t_0+T}(\mathbf{x})$  over time  $T$ . If a flow is defined on a uniform grid, the flow map deformation can then be computed by measuring the deformation of the regular configuration of the particles surrounding a central particle after transportation, as demonstrated by Figure 3. Given the flow map deformation matrix  $\mathbf{D}$ , the Cauchy Green deformation tensor can be computed as  $\nabla = \mathbf{D}^\top \mathbf{D}$ . The FTLE value is then [21]:

$$\sigma_{t_0}^T(x) = \frac{1}{|T|} \ln \sqrt{\lambda_{\max}(\nabla)} \quad (3)$$

**3.2.2. Attribute Accumulation.** Given the particles provided by the PFS data, it is important to identify those whose behaviors are of interest. One simple and intuitive approach is to inspect their overall (or average) behaviors over the course of simulation. Note that different quantities

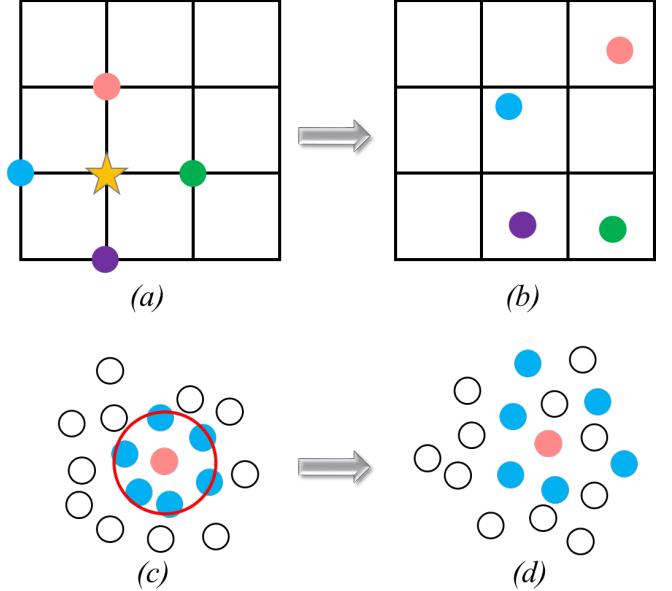


Figure 3. Top row shows the FTLE estimation for a flow defined on a regular grid. Star stands for the central position where the FTLE is of interest, while the colored dots indicate the starting positions of its directly neighboring particles (a) and their end positions over a time  $T$  (b). Bottom rows shows the FTLE estimation for a particle (red) in the PFS data. Blue particles are its neighboring particles. (c) shows their initial configuration, while (d) shows their end positions.

(or properties/attributes) (e.g., velocity, vorticity, density, etc.) can be carried by the particles. That said, the overall behaviors of the particles can be calculated by accumulating these quantities (or attributes) of interest over time. A similar accumulation framework has been proposed for integral curve classification [33] or the definition of an *attribute field* for flow segmentation and integral curve seeding [48], [50].

Consider a particle  $\mathbf{p}$  and its trajectory,  $\mathcal{C}_{\mathbf{p}}$ , that consists of the sequence of positions of  $\mathbf{p}$  over time. We accumulate a given attribute along  $\mathcal{C}_{\mathbf{p}}$  using  $\sum_{i=1}^n A(\mathbf{p}_i)$ , where  $A(\mathbf{p}_i)$  represents the attribute value associated with  $\mathbf{p}$  at its  $i^{\text{th}}$  position (or simulation step). We denote the accumulated attribute value by  $A(\mathbf{p})$ .

**3.2.3. Implementation.** Since traditional FTLE and attribute accumulation techniques could only work for rectilinear mesh-based data, we used Moving Least Square Fitting (MLSF) method for Lagrangian meshless framework.

Consider a given particle  $\mathbf{p}$  at time  $t_0$ . To estimate the flow map deformation centered at it over a time  $T$ , we first locate  $N$  direct neighboring particles of  $\mathbf{p}$ , denoted by  $\{\mathbf{q}_i\}$  (Figure 3 (c)). We then track their locations,  $\{\mathbf{q}'_i\}$  at  $t_0 + T$  (Figure 3 (d)). If the flow map deformation is  $\mathbf{D}$ , the following is satisfied

$$\mathbf{q}'_i - \mathbf{p}' = \mathbf{D}(\mathbf{q}_i - \mathbf{p}) \quad (4)$$

As the entries  $D_{ij}$  are unknown, estimating them can be achieved by solving the following least square fitting problem [1], [35].

$$\operatorname{argmin}_{\mathbf{D} \in \mathbb{R}^2} \sum_i^N \|(\mathbf{q}'_i - \mathbf{p}') - \mathbf{D}(\mathbf{q}_i - \mathbf{p})\|^2 \quad (5)$$

We use OpenGL and GLUI library to visualize and explore the PFS data generated by position-based fluid simulation. The basic visualization toolkit can be seen in Figure 4. We provide extensive user-friendly selection buttons to explore the domain which is of scientific interest. We would illuminate and discuss some findings in **Results and Discussions** part.

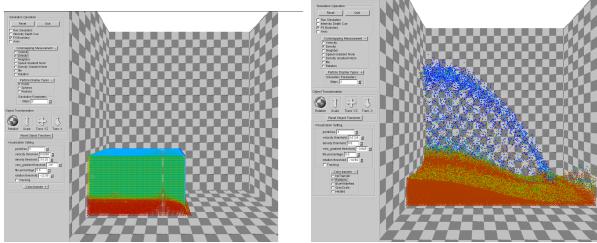


Figure 4. OpenGL visualization toolkit for fluid simulation data analysis(frame 250). We provide user interaction based on property of fluid data, and also by click **Run Simulation**, we can observe on-line fluid simulation demo.

## 4. Results and Discussions

Based on different property information conveyed by fluid particles, we're able to find some interesting results.

### 4.1. Particle Velocity

With given velocity threshold of 4.6974 in Figure 5, we found fewer than 7000 particles have velocity higher than this threshold, and they would splash up towards the boundary. The highest velocity particles always occur in the splash near the right boundary.

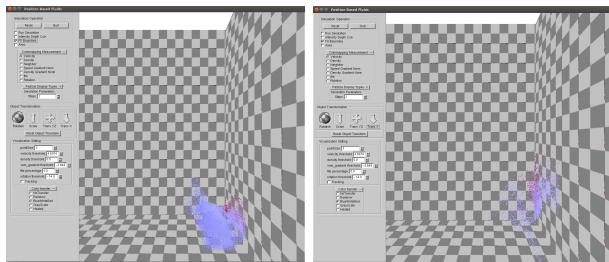


Figure 5. Simulation of particles whose velocity is above given threshold starting from frame 250(left) to frame 300(right). The complete particle distribution is as seen in Figure 4(right).

### 4.2. Particle Density

Since position-based fluid simulation obeys a density-constraint solution, the particle density equals rest density almost everywhere except on the surface and in the splash.

From color coding in Figure 6, we see particles all approach to rest density, and given a density threshold of 800, we found in frame 1000(Figure 6 (right)) nearly 98.95% can satisfy this condition.

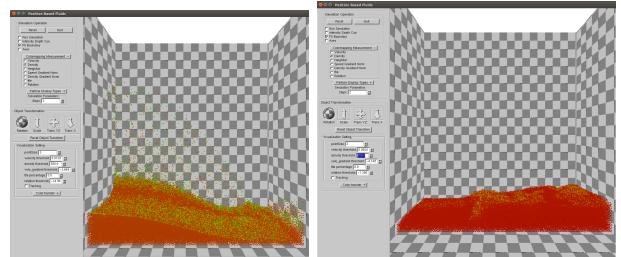


Figure 6. Particle density distribution on frame 250 (left) and 1000 (right). After splash cooling down, the fluid particles all tend to preserve rest density.

### 4.3. Particle Velocity Gradient

Taking velocity norm as a scalar value, we're able to use MLSF to get the spacial gradient of velocity norm. It's easier to be computed and visualized than Jacobian, because Jacobian has to involve tensor visualization. From Figure 7 we observe larger velocity norm gradient would either appear on splash or on surface wavelet.

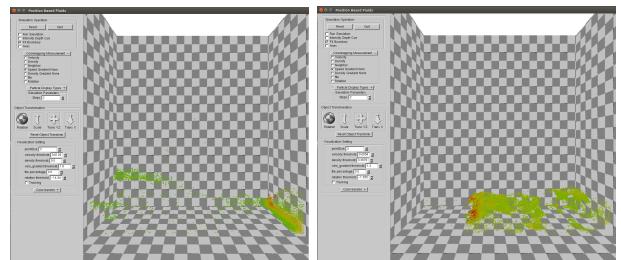


Figure 7. Particle distribution for gradient of velocity norm on frame 250 (left) and 1000 (right). Before splash, highest velocity norm gradient would appear on splash points. After splash cooling down, the highest velocity norm gradient tends to appear on surface of wave. The velo\_gradient threshold is symbolized by a log function.

### 4.4. Particle FTLE

FTLE is a very important feature for flow which characterizes the separation during physical domain along a specific time window size. In our computation, we use  $T = 50$  as time window size. We use threshold to extract particles with obviously higher ftle values in different frames as seen in Figure 8. We observe that largest FTLE either occur on top of splash, or somewhere hitting the boundary or the whirling on the surface wave. This is due to that only on these occasions does obvious flow seperation happen. Notice that in all our ftle threshold, we set the threshold to be 0.5, which means it should be more than half of the maximal ftle values.

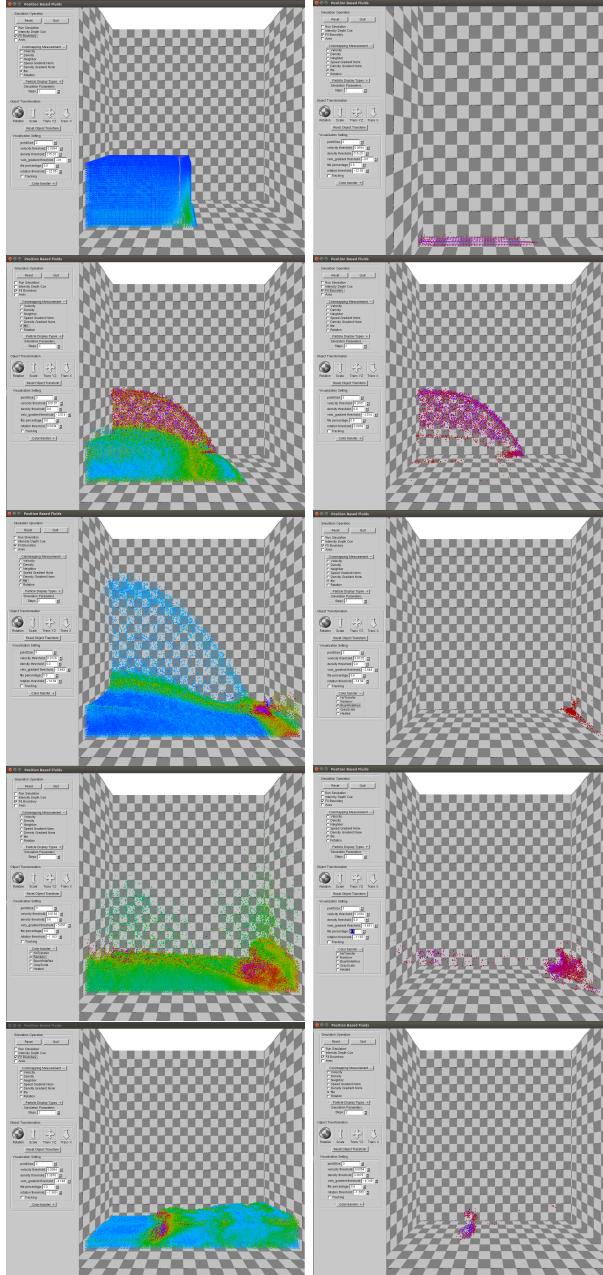


Figure 8. Particle distribution for a given ftle percentage threshold of original figures (left column) and extracted figures (right column). From top row to bottom row they're respectively frame 50, 150, 250, 500, and 1000.

Also we notice that ftle particle distribution might have some similarity with velocity-norm-gradient. In Figure ??, we found the similar distribution for particles with higher ftle and velocity-norm-gradient. However, it should be testified through further experiments, because due to neighbor insufficiency on the free flow surface, the moving least square fitting might have essentially erroneous estimate. We leave it for future work.

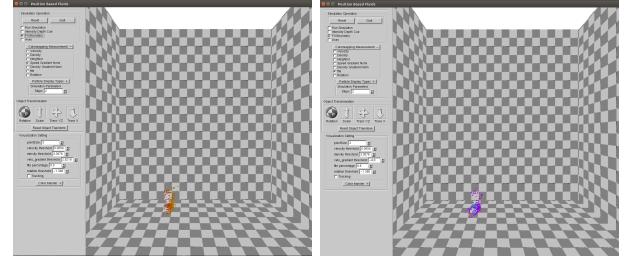


Figure 9. FTLE-threshold and velocity-norm-gradient threshold extraction figure for particles in frame 1000. We observe both largest velocity norm gradient and ftle occurs in the whirling part of surface wave.

#### 4.5. Particle Rotation

We would surmise that larger rotation values occur on the area close to boundary, especially on fluid-solid coupling bottom. This is because in these boundary areas, particles tend to have fierce and frequent velocity direction changes, which triggers larger rotation values. The Figure 10 verifies our deduction is almost right, but we can't explicate why this specific bottom boundary has this phenomena instead of other parts of bottom boundary.

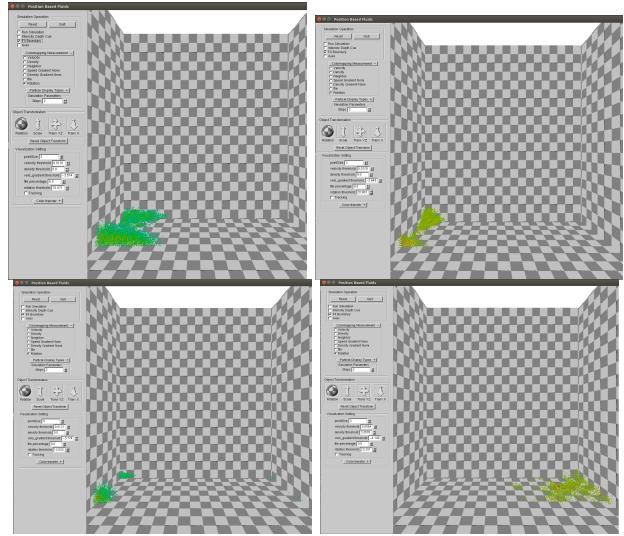


Figure 10. Rotation distribution for particles of each frame, respectively, 150, 250, 350, 1000. We observe all larger rotation areas are witnesses on the boundary. This is due to fierce velocity direction changes in boundary handling.

### 5. Conclusion and Future Work

We utilize an analysis-based framework for generated position-based fluid flow data, and perform FTLE and rotation computation for particle with moving least square fitting method. This framework enables Eulerian analysis technique to be applied in Lagrangian fluid simulation data, and enables user interactions for freely extract particles with adjustable threshold for particle property. We explore

the property extraction distribution with different frames of simulation, and found that the splash and surface wave has larger fte of interest, while bottom boundary area has larger rotation of interest. We're able to instruct domain experts to find anywhere in the fluid domain of interest based on extraction purpose.

However, there exists restriction for our framework in this paper. Firstly, we didn't visualize Jacobian with tensor technique, which is of greatest importance for fluid domain. Second, due to neighborhood insufficiency on the surface, the fte and Jacobian computation might be abnormal, and we look forward to more robust method to solve this problem. Third, our position-based fluid model can't preserve divergence-free constraint for incompressible fluid model, and so forth will have deviations in velocity-relative attribute estimate.

Our future work will mostly lie in several aspects besides solving the aforementioned limitations. Firstly, we wrote spherical and realistic representation in glui, but due to shortage of time we didn't implement that. We would try to use modern glsl and texture-based mothod to achieve a realistic visualization for fluid flow. Second, we would like to use highly parallizable CUDA program to accelerate computation of fluid simulation, which intends for an online computing and rendering procedure.

## Acknowledgments

The authors would like to thank Lei Zhang and Dr. Guoning Chen in DaViM Lab of University of Houston for the instruction and guidance in this work. We also would like to thank XXX for proofreading in this draft.

## References

- [1] A. Agranovsky, C. Garth, and K. I. Joy. Extracting flow structures using sparse particles. In *VMV*, pages 153–160, 2011.
- [2] A. Agranovsky, H. Obermaier, C. Garth, and K. I. Joy. A multi-resolution interpolation scheme for pathline based lagrangian flow representations. 2015.
- [3] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner. Versatile rigid-fluid coupling for incompressible sph. *ACM Trans. Graph.*, 31(4):62:1–62:8, July 2012.
- [4] M. Becker and M. Teschner. Weakly compressible SPH for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA ’07*, pages 209–217, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [5] J. Bender, M. Müller, and M. Macklin. Position-based simulation methods in computer graphics. In *EUROGRAPHICS 2015 Tutorials*. Eurographics Association, 2015.
- [6] J. Bender, M. Müller, M. A. Otraduy, M. Teschner, and M. Macklin. A survey on position-based simulation methods in computer graphics. *Computer Graphics Forum*, 33(6):228–251, 2014.
- [7] K. Bodin, C. Lacoursiere, and M. Servin. Constraint fluids. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):516–526, Mar. 2012.
- [8] I. Boxx, M. Stöhr, C. Carter, and W. Meier. Temporally resolved planar measurements of transient phenomena in a partially pre-mixed swirl flame in a gas turbine model combustor. *Combustion and Flame*, 157(8):1510–1525, 2010.
- [9] J. Chandler, H. Obermaier, and K. I. Joy. Interpolation-based pathline tracing in particle-based flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):68–80, Jan 2015.
- [10] Y. Chang, K. Bao, J. Zhu, and E. Wu. A particle-based method for granular flow simulation. *Science China Information Sciences*, 55(5):1062–1072, 2012.
- [11] J. E. G. Christiaan P. Gribble, Abraham J. Stephens and S. G. Parker. Visualizing particle-based simulation datasets on the desktop. *British HCI 2006 Workshop on Combining Visualization and Interaction to Facilitate Scientific Exploration and Discovery*, 2006.
- [12] L. Cullen and W. Dehnen. Inviscid smoothed particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 408(2):669–683, 2010.
- [13] L. F. d. S. Andrade, M. Sandim, F. Petronetto, P. Pagliosa, and A. Paiva. Sph fluids for viscous jet buckling. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 65–72, Aug 2014.
- [14] C. Deul, P. Charrier, and J. Bender. Position-based rigid body dynamics. *Computer Animation and Virtual Worlds*, 27(2):103–112, 2014.
- [15] M. Edmunds, R. S. Laramee, G. Chen, N. Max, E. Zhang, and C. Ware. Surface-based flow visualization. *Computers & Graphics*, 36(8):974–990, 2012.
- [16] J. Fang, A. Parriaux, M. Rentschler, and C. Ancey. Improved sph methods for simulating free surface flows of viscous fluids. *Applied Numerical Mathematics*, 59(2):251 – 271, 2009.
- [17] C. Garth, A. Wiebel, X. Tricoche, K. I. Joy, and G. Scheuermann. Lagrangian visualization of flow-embedded surface structures. *Computer Graphics Forum*, 27(3):1007–1014, 2008.
- [18] R. A. Gingold and J. Monaghan. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *mnras*, 181:375–389, Nov. 1977.
- [19] C. P. Gribble, T. Ize, A. Kensler, I. Wald, and S. G. Parker. A coherent grid traversal approach to visualizing particle-based simulation data. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):758–768, 2007.
- [20] T. Gnther and H. Theisel. Finite-time mass separation for comparative visualizations of inertial particles. *Computer Graphics Forum*, 34(3):471–480, 2015.
- [21] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena*, 147(3):352–370, 2000.
- [22] N. Kang and D. Sagong. Incompressible sph using the divergence-free condition. *Computer Graphics Forum*, 33(7):219–228, 2014.
- [23] B. Kohler, R. Gasteiger, U. Preim, H. Theisel, M. Gutberlet, and B. Preim. Semi-automatic vortex extraction in 4d pc-mri cardiac blood flow data using line predicates. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2773–2782, 2013.
- [24] R. S. Laramee, H. Hauser, L. Zhao, , and F. H. Post. Topology-based flow visualization, the state of the art. In H. H. H. Hauser and H. Theisel, editors, *Topology-Based Methods in Visualization 2005, Mathematics and Visualization*, pages 1–19. Springer-Verlag, 2007.
- [25] L. Lucy. A numerical approach to the testing of the fission hypothesis. *aj*, 82:1013–1024, Dec. 1977.
- [26] M. Macklin and M. Müller. Position based fluids. *ACM Trans. Graph.*, 32(4):104:1–104:12, July 2013.
- [27] J. Monaghan. Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics*, 44(1):323–346, 2012.
- [28] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992.

- [29] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [30] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, Apr. 2007.
- [31] X. Nie, L. Chen, and T. Xiang. An efficient sleepy algorithm for particle-based fluids. *International Journal of Computer Games Technology*, 2014:13, 2014.
- [32] D. I. Peter Horvath. Sph-based fluid simulation for special effects. *CESCG 2007*, 2007.
- [33] A. Pobitzer, A. Lez, K. Matkovic, and H. Hauser. A statistics-based dimension reduction of the space of path line attributes for interactive visual flow analysis. In *PacificVis*, pages 113–120, 2012.
- [34] B. Ren, X. Yan, T. Yang, C.-f. Li, M. C. Lin, and S.-m. Hu. Fast sph simulation for gaseous fluids. *The Visual Computer*, 32(4):523–534, 2016.
- [35] M. Robinson. Turbulence and viscous mixing using smoothed particle hydrodynamics. Ph.D. dissertation, Department of Mathematical Science, Monash University, 2009.
- [36] S. Shadden, F. Lekien, and J. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212(3–4):271–304, 2005.
- [37] X. Shao, Z. Zhou, and W. Wu. Particle-based simulation of bubbles in watersolid interaction. *Computer Animation and Virtual Worlds*, 23(5):477–487, 2012.
- [38] M. R. Sharen J Cummins. An sph projection method. *Journal of Computational Physics*, 152:584–607, 1999.
- [39] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible sph. *ACM Trans. Graph.*, 28(3):40:1–40:6, July 2009.
- [40] J. Staib, S. Grottel, and S. Gumhold. Visualization of particle-based data with transparency and ambient occlusion. *Computer Graphics Forum*, 34(3):151–160, 2015.
- [41] T. Takahashi, Y. Dobashi, I. Fujishiro, and T. Nishita. Volume preserving viscoelastic fluids with large deformations using position-based velocity corrections. *Vis. Comput.*, 32(1):57–66, Jan. 2016.
- [42] W. J. van der Laan, S. Green, and M. Sainz. Screen space fluid rendering with curvature flow. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, I3D '09, pages 91–98, New York, NY, USA, 2009. ACM.
- [43] T. Weaver and Z. Xiao. Fluid simulation by the smoothed particle hydrodynamics method: A survey. 2016.
- [44] H. Yan, Z. Wang, J. He, X. Chen, C. Wang, and Q. Peng. Real-time fluid simulation with adaptive SPH. *Comput. Animat. Virtual Worlds*, 20(2 - 3):417–426, June 2009.
- [45] J. Yu and G. Turk. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.*, 32(1):5:1–5:12, Feb. 2013.
- [46] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramee. Asymmetric tensor analysis for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):106–122, Jan. 2009.
- [47] F. Zhang, J. Wu, and X. Shen. Sph-based fluid simulation: A survey. In *Virtual Reality and Visualization (ICVRV), 2011 International Conference on*, pages 164–171, Nov 2011.
- [48] L. Zhang, R. S. Laramee, D. Thompson, A. Sescu, and G. Chen. Compute and Visualize Discontinuity Among Neighboring Integral Curves of 2D Vector Fields. In *Proceedings of TopoInVis*, Germany, 2015.
- [49] L. Zhang, R. S. Laramee, D. Thompson, A. Sescu, and G. Chen. Flow Visualization Based on A Derived Rotation Field. In *IS & T Visualization and Data Analysis Conference*, February 2016.
- [50] L. Zhang, R. S. Laramee, D. Thompson, A. Sescu, and G. Chen. An integral curve attribute based flow segmentation. *Journal of Visualization*, 19(3):423–436, 2016.
- [51] Y. Zhu and R. Bridson. Animating sand as a fluid. *ACM Trans. Graph.*, 24(3):965–972, July 2005.