Analysis Enhanced Particle-based Flow Visualization

Lieyu Shi, Lei Zhang, Wei Cao, Guoning Chen; University of Houston, Houston, Texas

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 accumlation 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.

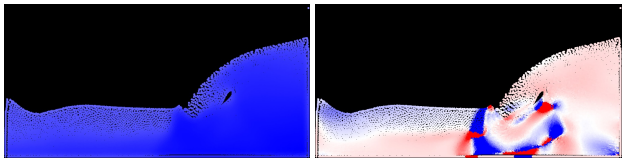

Figure 1: The direct rendering of particles of a 2D SPH simulation (left) and the visualization with different colors assigned to particles based on their characteristics (right).

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 [24]. Most of these features can not be revealed via direct rendering of the simulation results. The analysis and visualization of the 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 [25, 15] 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 [48, 32]. 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 [22] and Jacobian-based analysis , for the 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. Furthermore, based on these average and relative behaviors, we project the particles to an attribute space, with which the user can select the particles of interest to inspect intuitively. We have applied our comprehensive analysis and visualization framework to a number of 2D and 3D PFS data to demonstrate its effectiveness.

2 Related Work

Smoothed Particle Hydrodynamics (**SPH**) was first proposed by Gingold and Monaghan [19] and Lucy [26], and then it experienced extensive improvements and applications. Dependent on viscosity type, SPH can achieve good simulation results in highly viscous flow (like jet buckling [13], sand [55] and [10]) and inviscid flow [12]. Based on simulation scenarios, SPH can also be used to simulate gas [35], smoke [33] and bubbles [39]. There are tremendous applications and progress involved in SPH. We refer interested readers to a sequence of survey papers [29, 51, 28, 45] for a complete overview of this field.

Enforcing incompressibility is numerically challenging for SPH, because achieving divergence-free velocity field is computationally expensive and numerically unstable, especially with boundary conditions. Cummins et al. [40] 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 restricted, which hampers its application in practice. A prediction-correction incompressible SPH (PCISPH) was proposed by Solenthaler et al. [41], 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 [7, 3] or volume-preserving [43]. However, these methods still suffer from the limited time steps and can't balance between divergence-free and constant-density condition.

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 or velocities. It was first introduced to the graphics community by Muller et al. [31] to simulate rigid bodies and cloths, and later was 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. [6, 5] to acquire more details. Macklin al et. [27] used position-based method to achieve incompressiblity 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 [41]. We also refer readers to Fang et al. [16] which provides a rigid numerical method to distinguish surface particles. In PBF [27], particle anisotropy is firstly computed using anisotropic kernels [49], and then the fluid surface is constructed by the screen-space filtering technique [44].

Few efforts have been made on particle-based data analysis, and most of them are concerning visualization for large and time-varying particle datasets [11, 42, 20]. Chandler et al. [9] introduced an interpolation-based pathline tracing, but the computation is expensive and cannot provide physically intuive 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 [22, 38, 18, 21]. 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 square (MLS) for flow data from Eulerian simulations, but his work only focuses on 2D while in 3D both kernel searching and least square fitting for each particle is more complicated. Compared to the similar method proposed by Agranovsky et al. [1], our method directly applies to the particle based flow data without explicit neighborhood information for each particle, and with an isotropically dynamic radius, our method is robust to obtain enough neighboring particles for FTLE computation. Later, Agranovsky et al. [2] proposed a pathline-based representation for Lagrangian fluid flow data and reconstructs pathline surface, which has different goal from our work that aims to analyze and visualize particle-based simulated flow data.

3 Overview

Our framework includes two major components: 1) the particle-based fluid simulation (PFS) and 2) the analysis and visualization of the PFS results. The PFS is carried out using both SPH framework and the recently popular PBF framework[27]. The PFS results, including the position $\mathbf{p} = (x, y, z)$ and velocity $\mathbf{u} = (vx, vy, vz)$ at a given sampled time t for each particle, are saved into files. The analysis framework takes these files as input and performs the following tasks (see Figure 2):

1. Calculate the neighbors of each particle using a spatial binning;
2. Calculate the FTLE of each particle using the particle trajectory information;
3. Estimate the Jacobian (i.e., spatial gradient of the velocity field of the particles);
4. Calculate the accumulated attribute (e.g., rotation, curl, Jacobian determinant, etc.) associated with each particle and estimate the gradient of this accumulated attribute within a small neighborhood around the particle;
5. Apply transfer functions to visualize the analysis results and perform particle exploration to identify particles of interest based on their physical and derived attributes.

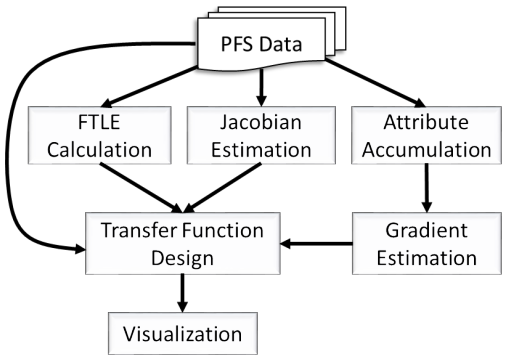

Figure 2: Analysis Framework

4 Particle-based Fluid Simulation

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 3 (left), enforcing compressibility highly depends on pressure computation, which is either achieved by weakly compressible SPH [30, 4] using

$$\begin{aligned} p &= k\rho \\ p &= k(\rho - \rho_0) \\ p &= B((\tfrac{\rho}{\rho_0})^{\gamma} - 1) \end{aligned} \tag{1}$$

or pressure projection for a divergence-free velocity field [40].

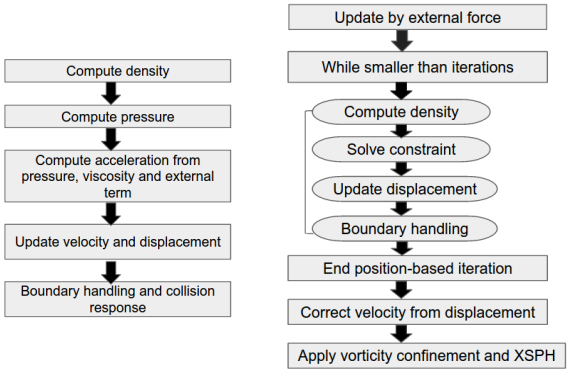

Figure 3: Pipeline comparisons between SPH (left) and position-based fluids (PBF) (right).

Position-based method (PBF) was firstly introduced into computer graphics community in [31] and then was applied to achieve incompressible fluid by enforcing the constant-density constraint in [27]. A prediction of particle displacement is iteratively updated by solving the non-linear density constraint function

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

The particle cohensiveness and numerical dissipation is improved by artificial pressure and vorticity confinement. The general pipeline is illustrated by the right flow chart in Figure 3.

In our practical experiement, we simulated the dam breaking scenarios with 2D SPH model, while for 3D flows we utilized the PBF framework. Since boundary handling in [27] that uses signed distance causes serious particle clumping, we adopted the boundary particle handling from [3] and relaxed density invariant method from [23] to improve the robustness in collision detection and simulation instability.

5 PFS Data Analysis

In this section, we describe a number of analysis techniques that help us better understand the behavior of the simulated free-surface fluids stemming from the above PFS solvers. As described earlier, the PFS simulation results are stored as the sequence of locations $\mathbf{p}$ of the individual particles over time. The velocity of each particle $\mathbf{u}$ at its given time is also stored. For the PFS result from the position-based fluid simulation, the vorticity information is also saved.

FTLE Computation

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 4. 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 [22]:

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

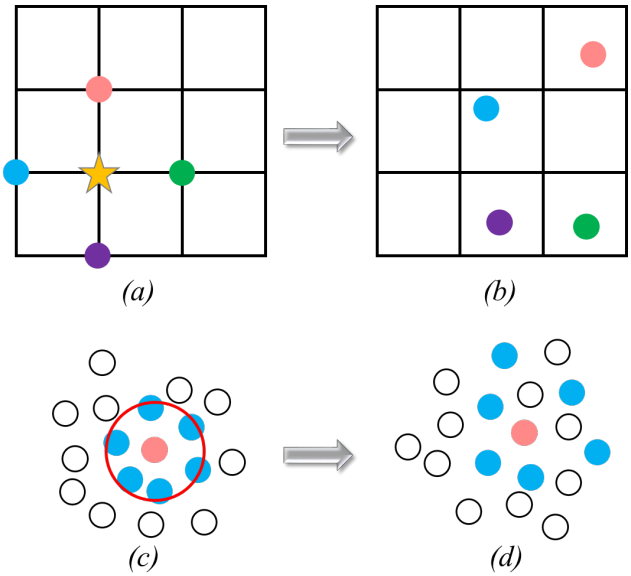

Figure 4: Top row shows the FTLE estimation for a flow defined on a regular grid. Star stands for the central position where the FTLE is of interest, while the colored dots indicate the starting positions of its directly neighboring particles (a) and their end positions over a time T (b). Bottom rows 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.

Based on the above description, the flow map deformation requires the assessment of the deformation of the configuration of neighboring particles. However, for PFS data sets, the neighboring particles are no longer regularly placed around the target position. Although a uniform grid may be imposed to the PFS data to utilize the above FTLE computation process, we opt for the estimation of the FTLE value associated with the individual particles directly because the flow map information is already provided by the particle positions over time. 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 4 (c)). We then track their locations, $\{\mathbf{q}_i'\}$ at $t_0 + T$ (Figure 4 (d)). If the flow map deformation is $\mathbf{D}$, the following is satisfied

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

As the entries D_{ij} are unknown, estimating them can be achieved by solving the following least square fitting problem [36, 1].

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

The above flow map deformation estimation with the least square fit can be sensitive to the neighborhood information. On the one hand, we need a sufficiently large neighborhood (e.g., to

cover at least 4 other particles in 2D or 15 other particles in 3D) in order to find a unique solution of Equation 5. On the other hand, larger number of particles may lead to slow computation. This asks for a proper neighborhood computation strategy, similar to the density estimation in the aforementioned particle-based simulation.

Due to unpredicted particle distributions, assigning a fixed neighborhood radius to all particles is sub-optimal for the MLS computation, e.g., a same radius for one particle might include more than 30 neighbors while for another particle it might only have a few neighbors. To address this, we propose an isotropic dynamic neighborhood searching based on the user-specified number. To be specific, given an initial value, the radius will be increased at a fixed speed unless the minimal number of neighbors are found(e.g., 15 particles in 3D in our experiments). This dynamic searching has also been applied in the neighborhood searching for mesh-based volume rendering .

Figure 5 compares the FTLE results of a Double Gyre flow [17] with $T = 10$. The left column shows the FTLE field of the flow computed using the conventional grid-based approach, while the right column shows the FTLE result using the above least square fitting approach. From the comparison, we see that the least square estimation returns quite similar FTLE field to the grid-based method.

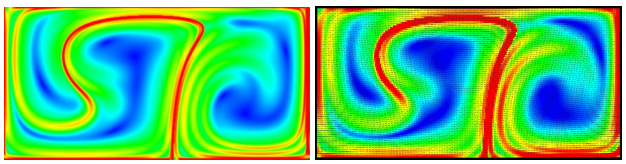

Figure 5: FTLE Results for Double Gyre defined on a uniform grid (left), and based on the advected particles using the proposed MLS approach (right).

Jacobian Estimation

For many flow data analysis tasks, such as vortex identification and asymmetric tensor field analysis [50], the Jacobian (i.e., the spatial gradient) of the velocity field of the flow needs to be computed. Similar to the above estimation of flow map deformation for the PFS data, we can estimate the Jacobian centered at a particle $\mathbf{p}$ at the given time by solving the following least square fitting problem, according to the linearization of vector field $L\mathbf{u}(\mathbf{x}+\Delta\mathbf{x}) = \mathbf{u}(\mathbf{x}) + \mathbf{J}\Delta\mathbf{x}$.

$$\underset{\mathbf{J}\in\mathbb{R}^2}{\operatorname{argmin}} \sum_{i}^{N} ||(\mathbf{u}_{\mathbf{q}_i} - \mathbf{u}_{\mathbf{p}}) - \mathbf{J}(\mathbf{q}_i - \mathbf{p})||^2 \tag{6}$$

where $\mathbf{u}_{\mathbf{q}_i}$ is the velocity vector associated with a neighboring particle $\mathbf{q}_i$ and $\mathbf{u}_{\mathbf{p}}$ is the velocity value at the central particle $\mathbf{p}$. After estimating the Jacobian associated with each particle over time, the Jacobian relevant characteristics can be derived (e.g., curl, divergence, vorticity, determinant, etc.), which are the attributes associated with the particle for the later processing.

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 (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 [34] or the definition of an *attribute field* for flow segmentation and integral curve seeding [52, 54].

Consider a particle $\mathbf{p}$ and its trajectory, $\mathscr{C}_{\mathbf{p}}$, that consists of the sequence of positions of $\mathbf{p}$ over time. We accumulate a given attribute along $\mathscr{C}_{\mathbf{p}}$ 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^{th} position (or simulation step). Note that this accumulation can start with any given time, say t_0, and carry out over time T, as illustrated by Figure 6. We denote the accumulated attribute value by $A(\breve{\mathbf{p}})$.

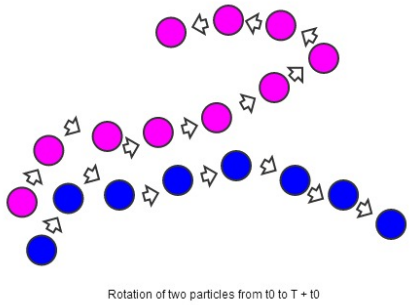

Figure 6: Illustration of the accumulation starting at t_0 till $t_0 + T$.

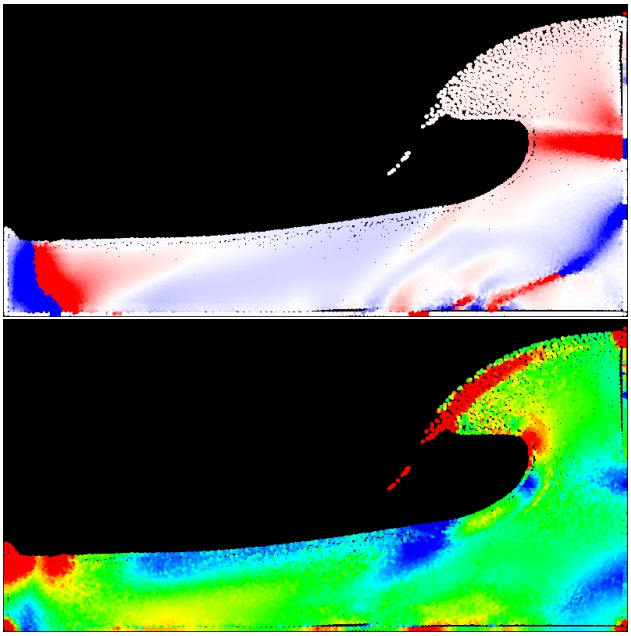

Figure 7: Top: the visualization of the accumulated rotation for a breaking dam simulation using a blue-white-red color scheme. Bottom: the visualization of the FTLE values with a rainbow color scheme.

Note that even though different local attribute values can be accumulated for the study of the overall behavior of the individual particles, we specifically focus on the accumulation of the rotation of the particle trajectories in this work. This is because this accumulated rotation information has high correlation to the vortices in the flow that are of interest to numerous applications. In addi-

tion, Zhang et al. [53] has shown that this accumulated rotation may also encode other flow characteristics, such as acceleration.

Figure 7 (top) visualizes the particles of a 2D breaking dam simulation based on their accumulated rotations. The rotation is estimated via the angle change of the velocity vector of the particles over time [53]. A blue-white-red color coding is used with red corresponding to positive (or counter clockwise) rotation and blue for negative (or clockwise) rotation. Compared to the FTLE result (bottom) that reveals separation behaviors of the flow near the surface, the accumulated rotation highlights some particles beneath the surface that may possess interesting rotation behavior.

Gradient of the accumulated attribute In many cases, important features (or boundaries of regions with different behaviors) can only be identified as the sharp change of certain quantity of interest between nearby samples. It is well known that the sharp change in a scalar field ϕ defined on a uniform grid can be captured via a standard gradient calculation, i.e., $\Delta\phi = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right)$

To estimate the gradient of an attribute of the particles ($\breve{A}$), we solve for the following least square fitting problem.

$$\underset{\breve{A}\in\mathbb{R}^2}{\operatorname{argmin}} \sum_{i}^{N} (< \mathbf{g}, \mathbf{q}_i - \mathbf{p} > -(\breve{A}(\mathbf{q}_i) - \breve{A}(\mathbf{p}))^2)^2 \tag{7}$$

where $\mathbf{g} = (g_x, g_y, g_z)$ is the gradient vector and $\breve{A}(\mathbf{q}_i)$ represents the attribute value at point $\mathbf{q}_i$. After estimating $\mathbf{g}$, we then compute and visualize its magnitude, $||\mathbf{g}||$. Figure 8 shows the accumulated rotation and its corresponding gradient magnitude of the Double Gyre flow. The design of the color transfer function is discussed in Section 6. From this result, we can see that the accumulated rotation along the trajectories of the individual particles tend to reveal additional information than the ridges of the FTLE field (Figure 5). This is because FTLE estimates flow map deformation based on only the end positions of the particles. Therefore, detailed intermediate information, such as sharp change of direction, along the trajectories can be neglected. Such sharp changes in the flow direction can indeed occur as the trajectories intersect with the paths of singularities (i.e., places where velocity value vanishes). This so-called *cusp-like* characteristic has been detected and utilized to improve the placement of pathlines [46].

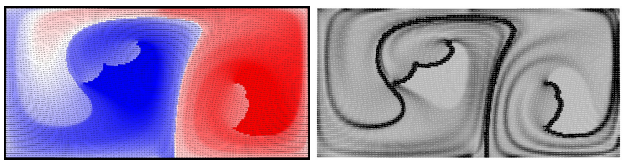

Figure 8: Visualization of the accumulated rotation (left) and its corresponding gradient (right) of the Double Gyre flow using the discrete estimation. A blue-white-red color scheme is used to color the individual particles based on their accumulated rotation values, while a gray scale color scheme is used to show the corresponding gradient magnitude.

6 Visualization and Exploration

To help experts and practitioners better understand and characterize their simulated PFS data, we impose a number of visual representations on the simulated particles based on their respective physical properties (e.g., velocity, vorticity and density) and the derived properties (e.g., FTLE value, Jacobian characteristics, and the accumulated attributes). In particular, we show this various information in two different ways. On the one hand, we generate color plots for 2D flow and particle/volume rendering for 3D data via a careful design of color transfer function (Section 6.1). On the other hand, we provide a simple visual analytic interface that enables the user to intuitively explore and select particles of interest based on various characteristics (e.g., their physical properties or the derived attributes) (Section 6.2).

6.1 Color Transfer Function

To map the quantities that particles are carrying to colors, the maximal and minimal values of these quantities typically need to be extracted. However, these two extrema values could fluctuate dramatically over time, making a simple linear color mapping based on them unstable (e.g., sharp color changes over time). Figure 9 demonstrates this visual discontinuity, in which the colors associated with the same particles have large change between two consecutive time steps due to the large fluctuation of the extrema values of the quantity. To address this issue, we map the original scalar value f using the following function.

$$f' = \begin{cases} f'_{max} & f > f'_{max} \\ f & \text{otherwise} \\ f'_{min} & f < f'_{min} \end{cases}$$

$$f'_{max} = \bar{f} + \alpha \times \sigma$$

$$f'_{min} = \bar{f} - \beta \times \sigma$$

where $\bar{f}$ is the average value of the given quantity at the given time which is relatively stable over time, and σ is the standard deviation. α and β are transfer parameters that can be changed in run time. In other words, one can alter the maximum f'_{max} and minimum f'_{min} thresholds by adjusting the values of α and β to achieve the desired visualization. After mapping the original scalar values to the new range, a linear color mapping can be performed in the range (f'_{min}, f'_{max}). In our experiments, a number of different color schemes are used, including the blue-white-red, gray scale and black-white-blue color schemes based on the above recalibration.

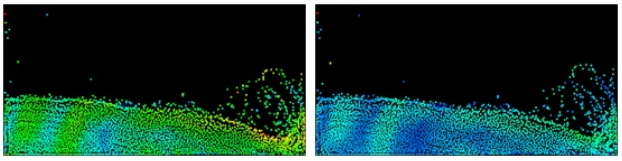

Figure 9: The color changes rapidly between two consecutive time steps without recalibrating the data range for color mapping.

6.2 Particle Exploration

For 2D simulations, a simple visualization with colors assigned to the individual particles based on their respective attributes may be sufficient, as shown in Figures 7, 12, 13, and 14. This is because 2D particles do not overlap. However, 3D PFS data sets do not have this property and their visualizations typically suffer from severe occlusion issue (Figure 15, top). Therefore, it is necessary to develop an exploration mechanism for the experts to highlight the particles whose behaviors are important,

while suppressing the other particles that are less relevant. To support this functionality, we devise a simple 2D attribute space so that the 3D particles can be projected onto this lower dimensional space for an easy exploration. Figure 10 provides an example of this 2D attribute space, in which the X axis represents the FTLE value and the Y axis is the accumulated rotation value. Specifically, Figure 10 (a) shows the scatter plot formed by the projected particles of a 3D simulation at step 399, which is heavily cluttering. To provide an overview of the distribution of the projected particles, we compute and visualize a 2D histogram by uniformly subdividing the attribute space, which provides better description on the distribution of the particles (Figure 10 (b)). A 50×50 2D histogram is used for all our results. With this 2D histogram, the user can easily select sets of particles for a detailed inspection. From the histogram shown in Figure 10 (b), we see that most particles have smaller FTLE and accumulated rotation values. The user can then focus on those small set of particles that have large FTLE or accumulated rotation values. More detailed discussions on the behaviors of the particles of this simulation are provided in Section 7.

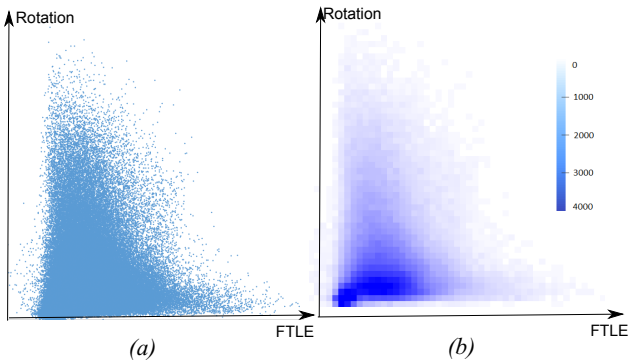

Figure 10: The scatter plot (a) defined by The FTLE and the accumulated rotation values of the particles. (b) shows the corresponding 2D histogram of the distribution of the projected particles.

7 Results

We have applied our analysis framework to a number of 2D and 3D PFS data sets. For the 2D PFS data sets, an SPH simulation was used, while the 3D PFS data sets were generated using a PBF simulation.

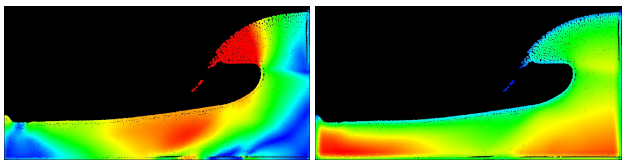
Figure 11: The velocity magnitude (left) and density (right) of a 2D breaking dam simulation with $\nu = 0.01$ at step 343.

For the 2D simulations, 6,000 particles were used. Figure 11 shows the velocity magnitude and density of a 2D breaking dam simulation with viscosity value $\nu = 0.01$. For both visualizations, a rainbow color coding is used. From this example, we see that in general, the particles tend to carry large velocity magnitude but small density in active areas, such as surface and splash. That is why people usually use this information to help the extraction of fluid surface and splash for rendering.

Comparison of fluids with different viscosities Figures 12, 13, and 14 provide the comparisons of FTLE, accumulated rotation and its gradient magnitude between fluids with different viscosities, respectively. From the comparisons, we see that the two simulations have generally similar overall patterns. On the one hand, the simulation with a smaller viscosity has richer and smaller-scaled fluid motions. For example, the fluid surface tends to have large perturbations (i.e., detailed splash) due to the smaller viscosity force. This characteristic is reflected by the richer patterns in its corresponding FTLE, rotation, and gradient visualizations. Specifically, the surface area of this simulation carries some large FTLE values. On the other hand, the simulation with a larger viscosity tends to have smoother fluid surface and little splash. Nonetheless, its underneath flow motions, as shown by our visualizations, are still rich but much quieter than those in the simulation with small viscosity, which is expected.

When comparing the results of FTLE and the accumulated rotation (and its gradient), one can see that the latter reveals more detailed and richer flow behaviors beneath the fluid surface. This may be because the fluid near the surface has larger distortion due to the pressure exerted by the waves and splash, while the disturbing caused by the waves and splash on the surface induces some rotating, turbulence like behaviors beneath the surface.

Particle selection for 3D PFS data For the 3D simulations, 128,000 particles were used. The computation of the FTLE, Jacobian estimation and attribute accumulation took about 2 second per frame. A kernel size is adaptively selected so that about 15 neighboring particles are used for the flow map deformation and Jacobian estimation. The time window size is $T = 40$ (i.e., 40 time steps). Figure 15 provides some particle exploration results of a 3D breaking dam simulation using a PBF solver. Time step 399 is shown here. The top row shows the direct particle rendering with a blue-white-color coding based on their respective attributes, i.e. FTLE (a) and accumulated rotation (b). For this incompressible fluid, the domain experts are typically interested in places where the flow has strong separation and rotation. While the direct rendering of particles assigned with colors based on their respective attributes may reveal this information well at the fluid surfaces, the underneath flow behaviors are hard to see due to the occlusion issue (top row of Figure 15). By selecting those particles of interest while suppressing the others, the places with strong separation and rotation in the flow are easily discernible (bottom row of Figure 15). From the results we see that particles with large FTLE values that indicate strong flow separation are located at or near the fluid surface (e.g., the front of a wave). Again, this can be explained by the large flow distortion (or stretching) caused by the force exterted by waves and splashes. In the meantime, the particles with large accumulated rotation are distributed near the bottom right of the simulation domain. This is partly because the particles at the lower right are about to hit the right wall and make a sharp change in their advection direction in a later time (Recall that we consider the particles in the next 40 time steps for the accumulation). Since we do not assume the no slip boundary condition in our simulations, those particles hitting the right wall are moving along it or bounced back. Combining with the pressure added by the upper particles, the turbulence-like behavior is induced beneath the surface, which is usually characterized

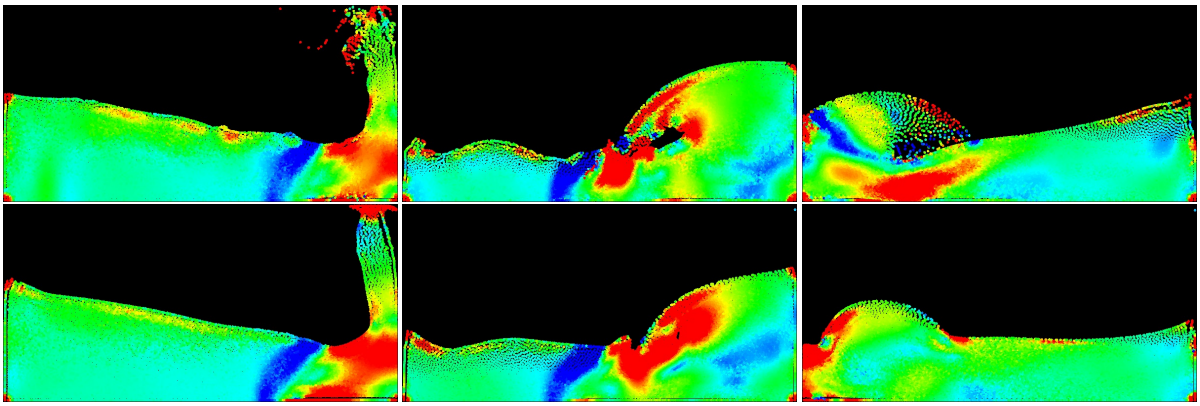

Figure 12: FTLE of two SPH simulations (Top: $\nu = 0.003$, bottom: $\nu = 0.01$)

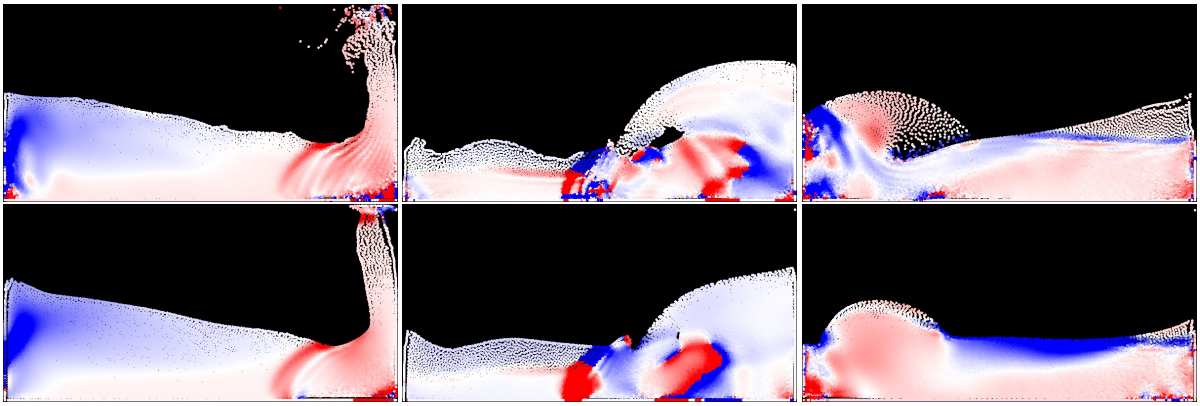

Figure 13: Accumulated rotation of two SPH simulations (Top: $\nu = 0.003$, bottom: $\nu = 0.01$)

by the strong rotation. Figure 16 shows the same simulation in a later time (i.e., at time step 430).

The 2D histogram (Figure 10 (b)) of the simulation shown in Figure 15 also reveals that particles that have large FTLE values typically do not have large accumulated rotation values. This is interesting, as it means that for this free surface fluid the strong flow separation near the surface is not associated with the strong rotation. In the meantime, the particles that have large accumulated rotation value (near the bottom) need not involve in flow separation.

Volume rendering of the 3D PFS data In addition to the rendering of particles with colors determined based on specific attributes, we also perform volume rendering based on the particle attributes. Specifically, we voxelize the simulation domain. For each voxel, we extrapolate the scalar value at its center based on the values associated with the nearby particles. Figure 17 shows the volume rendering results of a number of attributes extrapolated from the discrete particles. A $51 \times 122 \times 162$ voxelization is used. Compared to the direct rendering of discrete particles, volume rendering offers a smoother representation of the attributes, which may facilitates certain interpretation tasks, e.g., the identification of the ridges of the FTLE field. Nonetheless, due to its approximation nature, the volume rendering can only be used to provide an overview of the data.

8 Conclusion

In this paper, we presented an analysis and visualization framework for the particle-based fluid simulation (PFS) data. Compared to the analysis and visualization of flow data defined on fixed grid, PFS data does not provide the neighborhood information among particles, making their analysis with the conventional techniques difficult. To address this issue, we adapt a number of existing techniques for the mesh-based flow data to the PFS setting, including FTLE computation for the identification of the separation behaviors in the flow and the Jacobian based analysis for the identification of rotation and other kinematic motions in the flow. In addition, we adapt a recently introduced attribute accumulation framework for the characterization of the behaviors of the individual particles. A visual analytics interface is then designed to help experts explore various particle characteristics based on their accumulated attributes. We have applied our analysis framework to a number of 2D and 3D PFS data to demonstrate its utility. The results reveal rich particle behaviors (or motions) beneath the fluid surface, which has been ignored for free surface fluid simulation previously.

There are a number of directions we would like to explore in the future given the current preliminary, promising results for PFS data analysis. First, the particle selection/exploration framework can be improved by including the clustering technique introduced by Wu et al. [47] to provide the user with more support in particle selection. Also, supporting the selection of particles based on their multiple attributes at the same time will help experts better understand the correlation of various characteristics. Second, we plan to work with the domain experts to develop techniques based on the proposed framework to identify more complicated particle behaviors for their specific applications, e.g., the study of the erosion of the coastline caused by tsunami [37]. Third, our

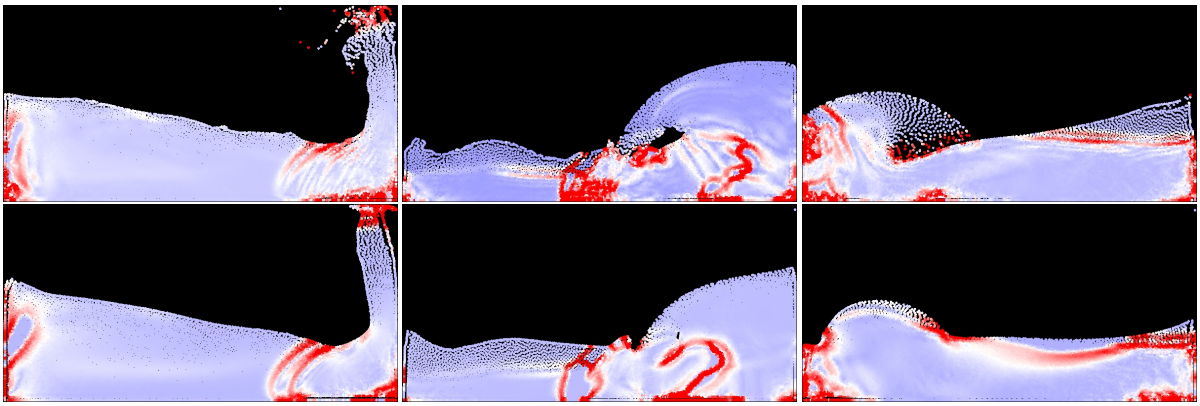

Figure 14: The gradient of the accumulated rotation of two SPH simulations (Top: $\nu = 0.003$, bottom: $\nu = 0.01$)

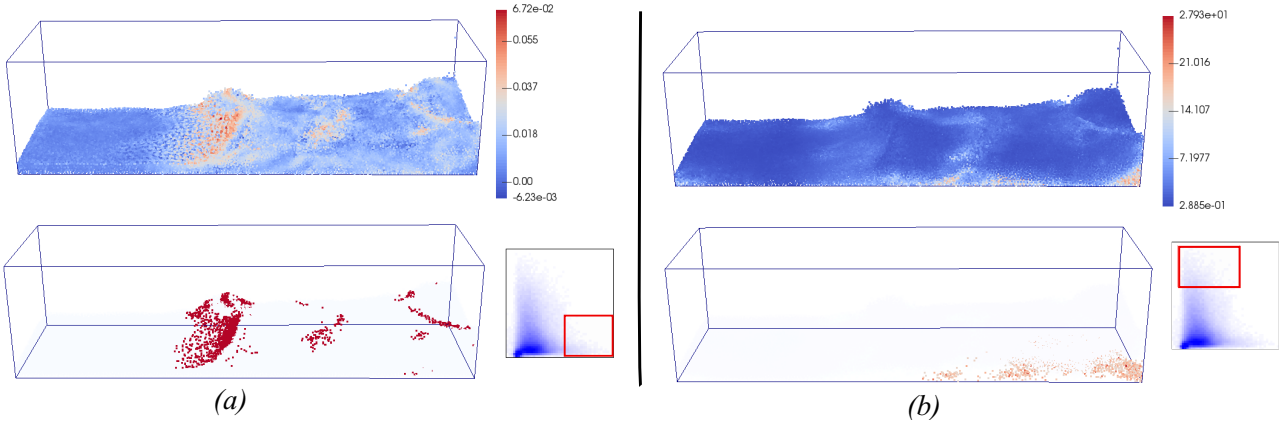

Figure 15: The visualizations of particles at time 399 based on their FTLE values (a) and their accumulated rotation values (b), respectively. Top row shows all the particles without selection, while the bottom row highlights those selected particles. The particles are selected based on the 2D histogram (shown in the inset).

analysis framework can be embedded into the simulation. Specifically, all the analysis discussed in this work can be carried out at any given time during the simulation based on the results from the preceding iterations. This can help experts re-configure their simulation during run time (i.e., in-situ) to allocate the computation power to places with particle dynamics that is critical to the specific applications.

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 multiresolution 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. Mller, M. A. Otaduy, 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.

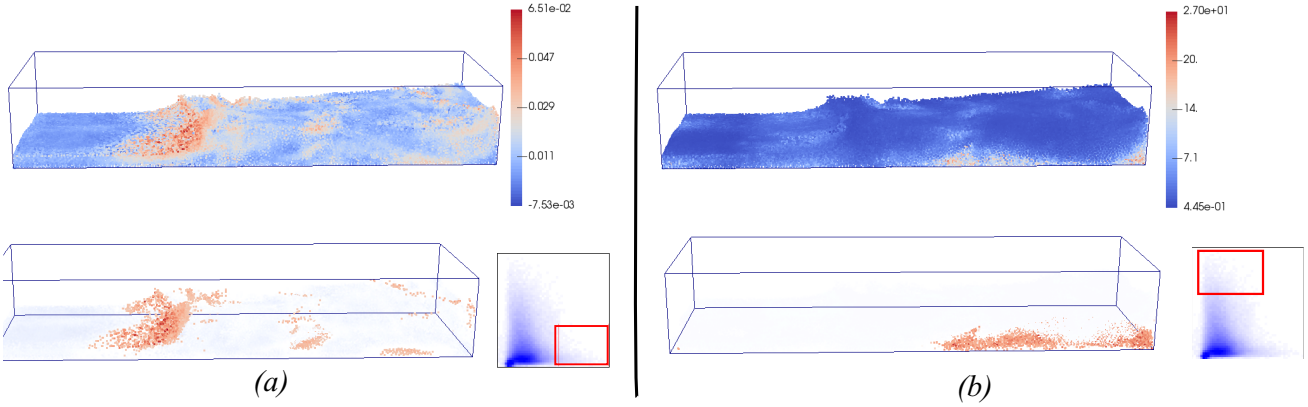

Figure 16: The visualizations of particles at time 430 based on their FTLE values (a) and their accumulated rotation values (b), respectively.

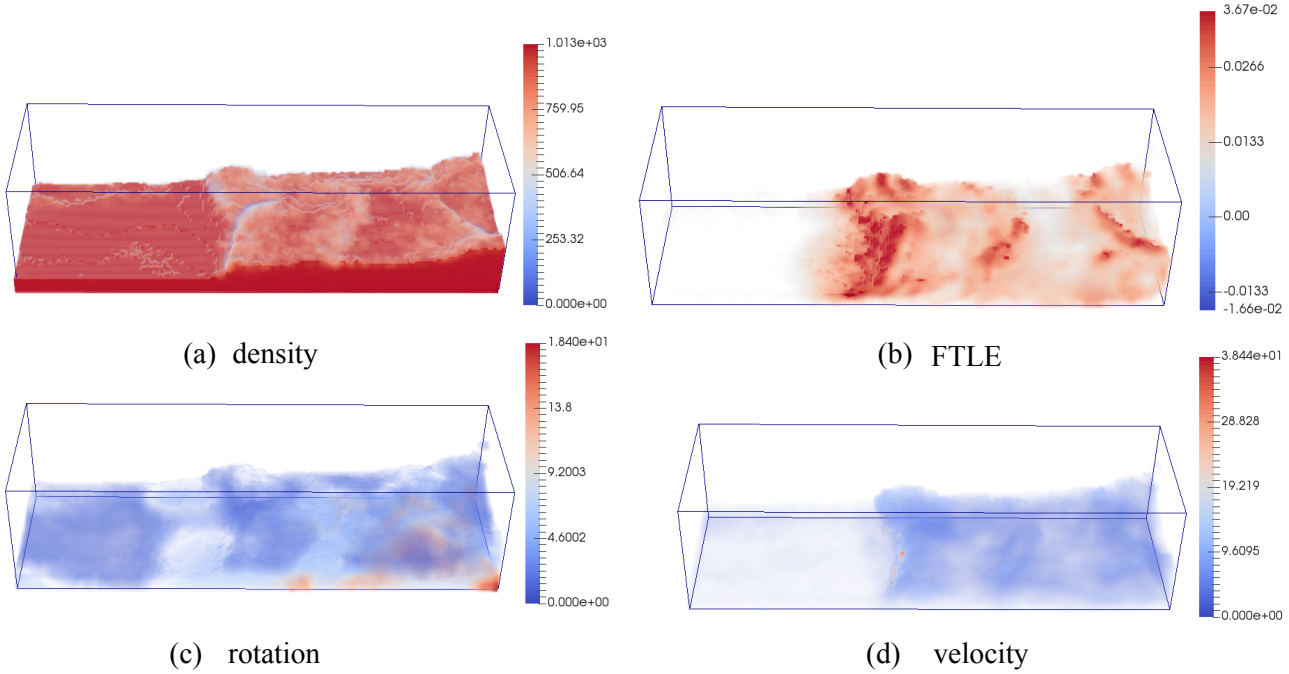

Figure 17: Volume renderings of the density (a), FTLE (b), accumulated rotation (c) and vorticity (d), respectively, at time 399 of a 3D PBF simulation.

[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] R. Fuchs, B. Schindler, and R. Peikert. Scale-space approaches to ftle ridges. In R. Peikert, H. Hauser, H. Carr, and R. Fuchs, editors, *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pages 283–296. Springer Berlin Heidelberg, 2012.

[18] 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.

[19] R. A. Gingold and J. Monaghan. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *mnras*, 181:375–389, Nov. 1977.

[20] 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.

[21] T. Gnther and H. Theisel. Finite-time mass separation for comparative visualizations of inertial particles. *Computer Graphics Forum*, 34(3):471–480, 2015.

[22] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena*, 147(3):352–370, 2000.

[23] N. Kang and D. Sagong. Incompressible sph using the divergence-free condition. *Computer Graphics Forum*, 33(7):219–228, 2014.

[24] 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.

[25] 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.

[26] L. Lucy. A numerical approach to the testing of the fission hypothesis. *aj*, 82:1013–1024, Dec. 1977.

[27] M. Macklin and M. Müller. Position based fluids. *ACM Trans. Graph.*, 32(4):104:1–104:12, July 2013.

[28] J. Monaghan. Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics*, 44(1):323–346, 2012.

[29] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992.

[30] 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.

[31] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, Apr. 2007.

[32] X. Nie, L. Chen, and T. Xiang. An efficient sleepy algorithm for particle-based fluids. *International Journal of Computer Games Technology*, 2014:13, 2014.

[33] D. I. Peter Horvath. Sph-based fluid simulation for special effects. *CESCG 2007*, 2007.

[34] 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.

[35] 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.

[36] M. Robinson. Turbulence and viscous mixing using smoothed particle hydrodynamics. Ph.D. dissertation, Department of Mathematical Science, Monash University, 2009.

[37] S. Sato, H. Yeh, M. Isobe, K. Mizuhashi, H. Aizawa, and H. Ashino. Coastal and nearshore behaviors of the 2011 tohoku tsunami along the central fukushima coast. In *Proc. Coastal Dynamics*, 2013.

[38] 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.

[39] 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.

[40] M. R. Sharen J Cummins. An sph projection method. *Journal of Computational Physics*, 152:584–607, 1999.

[41] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible sph. *ACM Trans. Graph.*, 28(3):40:1–40:6, July 2009.

[42] 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.

[43] 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.

[44] 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.

[45] T. Weaver and Z. Xiao. Fluid simulation by the smoothed particle hydrodynamics method: A survey. 2016.

[46] T. Weinkauf, H. Theisel, and O. Sorkine. Cusps of characteristic curves and intersection-aware visualization of path and streak lines. In *Topological Methods in Data Analysis and Visualization II*, pages 161–175. Springer, 2012.

[47] F. Wu, G. Chen, J. Huang, Y. Tao, and W. Chen. Easyxplorer: A flexible visual exploration approach for multivariate spatial data. *Computer Graphics Forum*, 34(7):163–172, 2015.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] Y. Zhu and R. Bridson. Animating sand as a fluid. *ACM Trans. Graph.*, 24(3):965–972, July 2005.