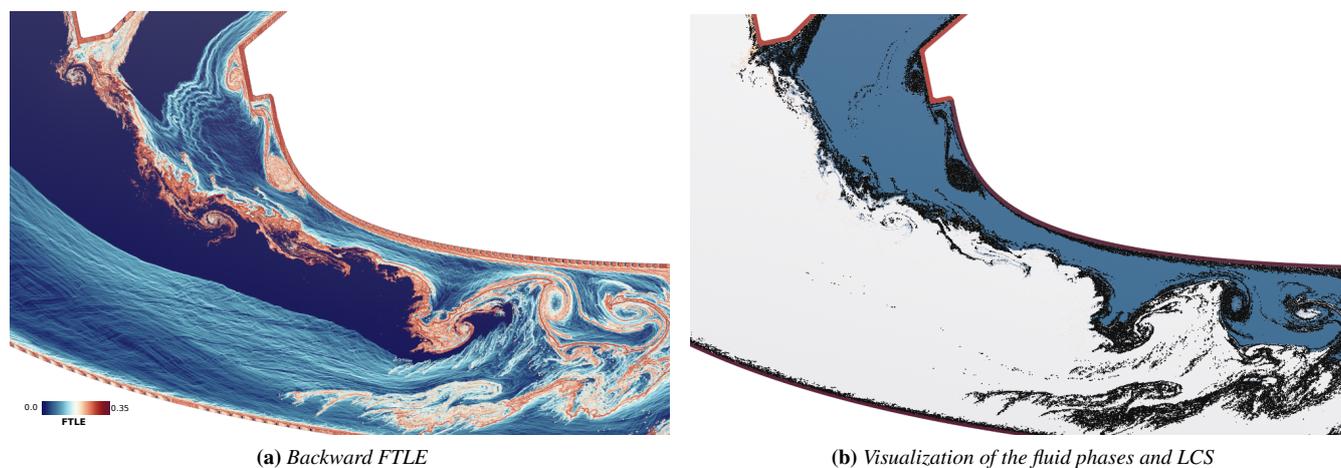


# Visualizing Transport and Mixing in Particle-based Fluid Flows

T. Rapp<sup>1</sup>  and C. Dachsbacher<sup>1</sup> 

<sup>1</sup>Karlsruhe Institute of Technology, Germany



**Figure 1:** We visualize time-dependent transport and mixing in a multiphase fluid simulation of a fuel spray nozzle. In (a), the backward FTLE shows the time-dependent flow behavior. In (b), the different fluid types are illustrated together with particles close to Lagrangian coherent structures (black).

## Abstract

To gain insight into large, time-dependent particle-based fluid flows, we visually analyze Lagrangian coherent structures (LCS), a robust skeleton of the underlying particle dynamics. To identify these coherent structures, we build on recent work that efficiently computes the finite-time Lyapunov exponent (FTLE) directly on particle data. We formulate the LCS definitions for particles based on robust approximations for higher-order derivatives of the FTLE. Based on these formulations, we derive a per-particle distance to the closest coherent structure. This allows us to visually analyze and explore the Lagrangian transport behavior directly on the particle data. We show that this is especially beneficial to detect and visualize flow features on different time scales. Lastly, we apply our approach to study mixing in multiphase flows by visualizing the different types of fluids and their relation to the coherent structures.

## CCS Concepts

• **Human-centered computing** → **Scientific visualization**;

## 1. Introduction

Interactive visualization of unsteady flows is still a big challenge in science and engineering. At the same time, simulations and the data they produce continue to grow rapidly. In recent years, particle-based simulation methods, e.g. the smoothed-particle hydrodynamics (SPH) method, have become popular. They are especially well-

suited for multiphase simulations that contain two or more distinct fluid phases, for example a liquid and a gas phase. In this work, we consider the visualization of such fluid flow datasets consisting of several million particles per time step.

The finite-time Lyapunov exponent (FTLE), a measure for the rate of separation (or attraction) of infinitesimally close tracer par-

ticles, has been established as a powerful way to visualize time-dependent flows. Recent work computes the FTLE directly on particle data [SCMZ16, SZCC17, DRC\*18]. The otherwise costly FTLE computation is thereby considerably simplified due to the inherent Lagrangian nature of particle data. The FTLE can be employed to extract Lagrangian coherent structures (LCS), a robust skeleton of Lagrangian particle dynamics that organize fluid transport and mixing behavior [Hal15]. Although a significant amount of research has been conducted on extracting Lagrangian coherent structures from FTLE fields, this has been mostly limited to structured data. It is possible to resample particles to a grid, but this reduces the effective resolution and amplifies the error stemming from the approximation of the first and second derivatives of the FTLE field, which are needed for the extraction of LCS. In this work, we consider the evaluation of LCS directly on particle data, for which robust approximations for the derivatives exist. Furthermore, we reformulate the LCS criteria for particles and their local neighborhood to derive a per-particle distance to the closest LCS.

We further discuss the visual analysis of large, multivariate flows based on this LCS distance to visualize the Lagrangian particle dynamics and to explore the correlations to other flow features. To reduce visual clutter in large particle datasets, finding and extracting features of interest is especially important. We show that our derived LCS distance is well-suited in highlighting relevant structures of the flow. Furthermore, we facilitate the interactive exploration and analysis of multivariate particle data and derived features using multiple coordinated views, including a parallel coordinate plot, histograms, and scatter plots of large particle datasets. Using the concept of brushing and linking, the user can interactively explore and analyze the multivariate data and derived features, such as the correlation to LCS. The potential of combining visual analysis and feature extraction has already been demonstrated in previous work [BMDH07, STH\*09, LZM\*11]. Furthermore, our findings indicate that the visual analysis based on Lagrangian coherent structures is an effective approach to study the transport behavior in time-dependent flows.

The selection of the time scale greatly influences the amount and type of features that are found in the flow. With a fast evaluation of LCS and the user interaction in all of the linked views, the identification and comparison of appropriate time scales is significantly improved. Lastly, we apply our approach to visualize mixing in multiphase fluid flows. Since LCS act as transport barriers that are minimally diffusive, they are closely related to the interface between two phases, e.g. liquid and gas particles, and their mixing properties [MHP\*07]. By correlating the phase interface with the LCS, we find an effective visualization of the mixing behavior in multiphase flows.

To summarize, our main contributions are:

- We derive a formulation for Lagrangian coherent structures directly on particle-based flows,
- We discuss the visual analysis of Lagrangian transport based on the LCS,
- We identify LCS to visualize mixing behavior in multiphase flows.

## 2. Related Work

This section discusses prior studies related to the visualization of particle-based flows and introduces Lagrangian coherent structures.

### 2.1. Lagrangian Coherent Structures

Lagrangian coherent structures are the most repelling, attracting, and shearing material surfaces in the flow. As such, they characterize the topology of the flow and govern the global transport behavior, see Haller [Hal15] for a recent overview. In order to identify LCS, the finite-time Lyapunov exponent has been proposed in the seminal work by Haller [HY00]. This scalar measure describes the rate of separation (or attraction) with respect to infinitesimally close particles over a finite-time interval. More specifically, let the flow map  $\phi_{t_0}^t(x)$  be the mapping from a position  $x$  of a particle at time  $t_0$  to its position at time  $t$ . With the spatial gradient  $\nabla\phi_{t_0}^t(x)$ , the *right Cauchy-Green strain tensor* is defined as

$$\mathbb{C}(x, t_0, t) := \nabla\phi_{t_0}^t(x)^\top \nabla\phi_{t_0}^t(x). \quad (1)$$

Using the largest eigenvalue  $\lambda_{\max}$  of  $\mathbb{C}$ , the FTLE is defined as:

$$\sigma(x, t_0, t) := \frac{1}{|t - t_0|} \ln \left( \sqrt{\lambda_{\max}(\mathbb{C}(x, t_0, t))} \right). \quad (2)$$

The FTLE thus describes the average exponential stretching of an infinitesimally close particle neighborhood at time  $t_0$  when the flow is integrated to  $t$ . The FTLE that is computed by integrating forward in time from  $t_0$  to  $t$  consequently measures the rate of separation. By integrating backward in time, the rate of attraction is computed instead, which is referred to as the backward FTLE.

There has been a lot of research in the computation of the FTLE to reduce its computational effort [BR10, SRP11, BGT12, CRK12]. Most of the effort in the computation of the FTLE stems from the dense integration of tracer particles that is required to approximate the spatial gradient  $\nabla\phi_{t_0}^t(x)$ . For Lagrangian flows, this integration can be replaced by computing the derivative from existing pathlines. More specifically, Agranovsky et al. [AGJ11] integrate a sparse set of particles and employ moving least squares to compute the derivatives necessary for the computation of FTLE. Shi et al. [SZCC17] similarly solve a least-square fitting problem to compute FTLE. Sun et al. [SCMZ16] derive a SPH formulation for the FTLE computation on particles, where the derivatives are obtained by deriving the smoothing kernels. Dauch et al. [DRC\*18] discuss the efficient implementation of this formulation on the GPU to interactively compute the FTLE for large, three-dimensional datasets.

Given an FTLE field, locally maximizing surfaces in at least one dimension, referred to as height ridges, can be defined as Lagrangian coherent structures [Hal01, SLM05]. Sadlo and Peikert [SP07] present a method to efficiently extract and filter height ridges using marching ridges and adaptive mesh refinement. Garth et al. [GGTH07] propose an efficient approximation of the FTLE field, but prefer a direct visualization instead of ridge extraction. The authors discuss the challenge of ridge extraction from intrinsically noisy FTLE fields. More recent work [Hal11] shows deficiencies of the definition of LCS as height ridges and instead introduces the notion of weak LCS. This definition is not only based on the FTLE and its derivatives, but directly on the strain tensor  $\mathbb{C}$ . Schindler et al. [SPFT12] propose a ridge concept based on the weak LCS

definition, which they call *C-Ridges*. In this work, we consider both LCS definitions. An alternative approach to approximate Lagrangian features for two-dimensional flows is proposed by Kuhn et al. [KER\*14], based on a geometric reconstruction of the flow map.

## 2.2. Particle-Based Flow Visualization

Krüger et al. [KKKW05] and Bürger et al. [BSK\*07] present particle tracing systems that interactively integrate particles in a vector field and display them using a glyph-based representation. Bürger et al. [BKKW08] propose an importance measure to reveal important structures of flows and further employ FTLE to place so called anchor lines. Jones et al. [JMEL08] visualize multivariate particle data using transparent glyphs and pathlines. The authors emphasize the advantages of parallel coordinates for visually analyzing particle data. To visually debug SPH simulations, Reinhardt et al. [RHD\*17] similarly combine a spatial 3D view with additional views, such as scatter and parallel coordinate plots. As shown by Grottel et al. [GKM\*15], the visualization of particle data benefits from specific data structures and visualizations.

Salzbrunn et al. [SGSM08] propose pathline predicates, which filter a set of pathlines according to properties that are of interest to the user. Shi et al. [STH\*09] analyze unsteady 3D flow fields by computing multiple properties of selected pathlines. The resulting multivariate data is analyzed using information visualization approaches with a set of linked views, including brushing and a focus+context visualization. To visualize particle data, Shi et al. [SZCC17] compute the FTLE and further accumulate pathline attributes on particle data, but do not consider the evaluation of LCS.

## 3. Lagrangian Coherent Structures in Particle-based Flows

In this section, we first detail how to compute spatial derivatives on particle data and how this can be used to compute the FTLE. Then, we introduce the height-ridge and weak LCS definitions. Afterwards, we recapitulate the extraction of LCS from an FTLE field defined on a uniform grid. Finally, we discuss the evaluation of LCS on particles.

### 3.1. Spatial Derivatives and FTLE

The computation of FTLE and LCS requires the evaluation of higher-order spatial derivatives. For particle data, moving least squares [LS81] can be applied to reconstruct a smooth and continuous fit to the particle data. It has been successfully applied to compute higher order derivatives [CFCN\*07], in which case a basis of at least quadratic degree is required. Agranovsky et al. [AGJ11] employ moving least squares for the computation of FTLE on sparsely integrated tracer particles.

For our experiments, we follow recent work for SPH data [SCMZ16, DRC\*18], which we discuss in more detail. To obtain spatial derivatives in the SPH framework [Mon92] in  $n$ -dimensional space, we first define the interpolation of a quantity  $A_i$  at position  $x_i \in \mathbb{R}^n$  using the neighboring particle positions  $x_j \in \mathbb{R}^n$ :

$$A_i = \sum_j A_j W(x_j - x_i, h), \quad (3)$$

where  $W$  is a compact kernel function parameterized by the smoothing length  $h$ . The neighboring particle positions  $x_j \in \mathcal{N}(x_i)$ , where  $\mathcal{N}(x_i) \subset \mathbb{R}^n$ , thus depend on the support of the kernel function. To compute the gradient, the kernel function can be differentiated:

$$\nabla A_i = \sum_j A_j \nabla W(x_j - x_i, h). \quad (4)$$

Although it is possible to compute second derivatives the same way, this is discouraged. Instead, we use the following approximation [Bro85]:

$$\nabla^2 A_i = \sum_j (A_j - A_i) \frac{(x_j - x_i)^T \nabla W(x_j - x_i, h)}{\|x_j - x_i\|}. \quad (5)$$

To compute the FTLE at position  $x_i$ , we consider how the neighborhood  $\mathcal{N}(x_i)$  at time  $t_0$  is deformed at time  $t$ , which leads to the following definition for the gradient of the flow map [SCMZ16]:

$$\nabla \phi_{t_0}^t(x_{i,t_0}) = \sum_j (x_{j,t} - x_{i,t}) \otimes \nabla W(x_{i,t_0} - x_{j,t_0}, h), \quad (6)$$

where we index the positions by time. Similar to Dauch et al. [DRC\*18], we employ a correction term for the FTLE computation due to the particle disarray [BL99].

### 3.2. Definition of LCS

We discuss two different LCS definitions: The first one defines LCS as height ridges of the FTLE, while the second one is based on the concept of weak LCS as defined by Haller [Hal11].

To detect LCS as  $(n-1)$ -dimensional height ridges within the  $n$ -dimensional FTLE field  $\sigma$  at position  $x$ , we compute the gradient  $\nabla \sigma(x)$ , the Hessian matrix  $H_\sigma(x)$ , the smallest eigenvalue  $e_{\min}$  and corresponding eigenvector  $v_{\min}$  of  $H_\sigma(x)$ . The following two criteria must hold:

$$c_1^h(x) := \nabla \sigma(x) \cdot v_{\min} = 0, \quad (7)$$

$$c_2^h(x) := e_{\min} \leq 0. \quad (8)$$

Note that we relax the inequality in the second criterion to include thick ridges, similar to Farazmand and Haller [FH12].

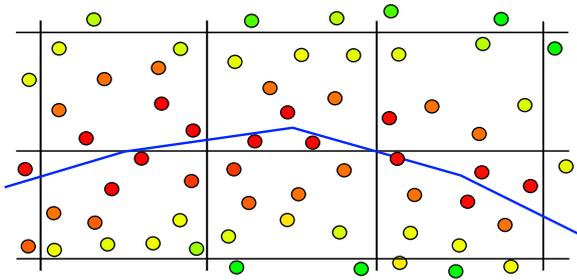
In several examples, Haller [Hal11] shows that observable LCS are not necessarily FTLE ridges and vice versa. We also use the more accurate definition of weak LCS, but reformulate it to fit into two criteria. To this end, we require the strain tensor  $\mathbb{C}(x, t_0, t)$  that is used to compute the FTLE. We compute eigenvalues  $\lambda_i$  of  $\mathbb{C}(x, t_0, t)$ , with  $\lambda_1 \leq \dots \leq \lambda_i \leq \dots \leq \lambda_n$ , and the corresponding eigenvectors  $\xi_i$ . Since the major eigenvector  $\xi_n$  of  $\mathbb{C}(x, t_0, t)$  is equal to the direction of maximal stretching, it must be orthogonal to the LCS:

$$c_1^w(x) := \langle \nabla \lambda_n, \xi_n \rangle = 0. \quad (9)$$

For the second criterion, we reformulate Equation 8 since the LCS must be a local maximum in direction  $\xi_n$ . An additional condition ensures that the normal repulsion rate is larger than the tangential stretch, typically caused by shearing in the flow. To summarize, we compute the second criterion as:

$$c_2^w(x) := \langle \xi_n, \nabla^2 \lambda_n \xi_n \rangle \leq 0 \wedge \lambda_{n-1} \neq \lambda_n \wedge \lambda_n > 1. \quad (10)$$

With these criteria, we fulfill three of the conditions necessary for



**Figure 2:** Illustration of the per-particle LCS distance (increasing from red to green over yellow), and the corresponding LCS created by marching ridges (blue).

a weak LCS (cf. Haller [Hal11], theorem 7). Note that the height ridge definition only fulfills two of those conditions, but does not depend on the strain tensor. We can make use of both definitions and thus refer to the criteria simply as  $c_1$  and  $c_2$ .

### 3.3. Grid Extraction

For grid-based FTLE fields in 3D, the extraction of LCS as height ridges of the FTLE can be performed with the Marching Ridges algorithm [FP01] as discussed by Sadlo and Peikert [SP07, SP09]. In detail, the locations where the first criteria  $c_1$  are met is found using Marching Cubes [LC87], i.e. computed at the grid vertices and approximated using linear interpolation inside of each grid cell. Note that the orientation of eigenvectors in a cell must be made consistent, which can be performed with principal component analysis (PCA). Afterwards, a triangle is created only if the second criterion  $c_2$  is met for the corresponding edge. Noise amplification due to the use of second derivatives is addressed by applying smoothing prior to the evaluation of derivatives. Additionally, feature filtering is applied since the approach often yields more ridges than desired, for example due to degeneracies of the Hessian [STS10].

### 3.4. LCS from Particles

Performing Marching Ridges on particle data would require resampling the FTLE to a grid and computing the derivatives afterwards. This step is computationally inefficient and is prone to amplify noise in the derivatives. Instead, we compute the derivatives and evaluate the LCS criteria for each particle using its local neighborhood. Similar to Marching Ridges, we compute zero crossings of the first criteria from a particle with respect to its neighboring particles; however, we cannot triangulate the zero crossings. Instead, we compute the minimal distance to any LCS in its neighborhood for each particle, cf. Figure 2. In the following, we discuss this procedure in detail.

Given a particle at position  $x \in \mathbb{R}^n$  and neighboring particles  $x_j \in \mathcal{N}(x)$ , we compute the distance from  $x$  to the nearest LCS in the neighborhood. First, we make the orientation of eigenvectors for the particle and its neighbors consistent using PCA. Afterwards, we check each neighboring particle if  $c_1(x_j)$  has a different sign than  $c_1(x)$ . In this case, the position of the zero crossing LCS( $x, x_j$ ) between the two particles can be determined using linear interpolation. Finally, if the zero crossing exists, and both particles additionally

fulfill criterion  $c_2$ , we obtain the distance to  $x$  as:

$$d_{\text{LCS}}(x, x_j) := \begin{cases} \infty & \text{if } \neg (c_2(x) \wedge c_2(x_j)), \\ \infty & \text{if } \text{sign}(c_1(x)) = \text{sign}(c_1(x_j)), \\ \|\text{LCS}(x, x_j) - x\| & \text{otherwise.} \end{cases} \quad (11)$$

We define the LCS distance for  $x$  as the minimal distance to the LCS towards all neighboring particles:

$$d(x) := \inf_{x_j \in \mathcal{N}(x)} d_{\text{LCS}}(x, x_j). \quad (12)$$

This function defines an unsigned distance field. Even though we can compute points on the LCS using linear interpolation, an extraction of a surface is difficult since the distances are unsigned and the LCS contain complicated, nonorientable structures. We did perform several experiments to extract the coherent structures, but were unable to find a reliable method. However, we did not consider the method by Kindlmann et al. [KCH\*18], which seems promising.

Instead of extracting surfaces, we store for each particle the distance to the closest LCS. These per-particle distances can be added to the dataset and used for further analysis and visualization. Lastly, we normalize the distance using the radius  $r$  of the particle neighborhood:

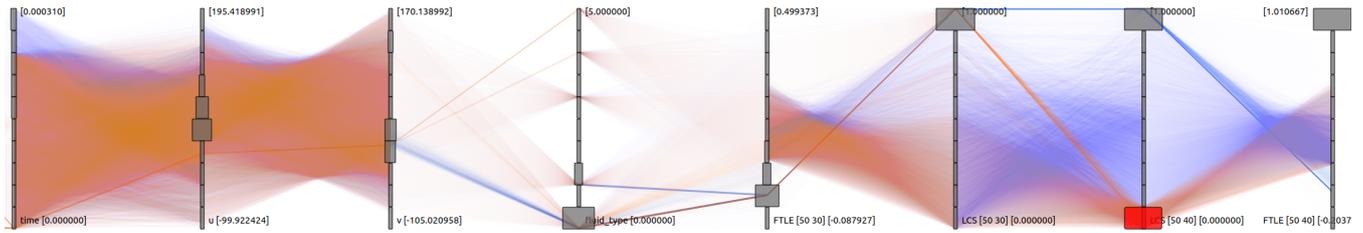
$$\mathcal{L}(x) := \frac{d(x)}{r}. \quad (13)$$

## 4. Particle-Based Flow Visualization

For an effective particle-based flow visualization, the amount of visual occlusion has to be reduced to make important parts of the flow visible. This is crucial for particle data where datasets contain millions of particles, most of which are not of immediate interest to the user. We employ the FTLE, LCS distance, and the multivariate particle data to select particles of interest. In the following, we describe the visual analysis in multiple linked views. Then, we consider the visualization of particle trajectories. Lastly, we discuss the visualization of mixing in multiphase fluid flows.

### 4.1. Visual Analysis

In addition to a spatial visualization of the particles, we employ multiple linked views on the data. Interesting particles can be selected by brushing in any of the views to focus on the currently deemed relevant parts of the flow. Central for the visual analysis is the parallel coordinate plot that shows all particle attributes, including the position and velocity, but also other derived quantities such as vorticity, FTLE, and LCS distance. For each of those dimensions an axis is created and for every particle a polyline is drawn over all axes based on the particle's values. The resulting plot allows correlating each pair of neighboring dimensions. To this end, the axes can be reordered to compare different dimensions. Moreover, every axis can be brushed to select particles. The polylines of selected particles are highlighted in the parallel coordinate plot and in all of the other linked views. Correspondingly, the spatial visualization shows only selected particles, which are affected by the transfer function. Figure 3 depicts a parallel coordinate plot, where a low LCS distance has been brushed. Consequently, particles near a LCS are selected and we can immediately see how these particles are correlated to and distributed in other dimensions.



**Figure 3:** The parallel coordinate plot shows all multivariate dimensions of each particle at once. Here, particles close to a LCS are brushed (red) on the corresponding axis. The brushed lines, highlighted in orange, allows correlating particles near a LCS to the other dimensions, for example to the FTLE (on the right of the brushed axis) or to LCS over a different time interval (on the left).

For large particle datasets, drawing millions of polylines quickly leads to clutter. We depict a 1D histogram on each axis, which helps to convey the particle distribution in each dimension. Moreover, we employ a frequency-based approach [NH06, HW09]. Instead of drawing single polylines, the line density is computed and mapped to color and transparency. As proposed by Guo et. al. [GXY11], we employ a non-linear mapping of density to avoid losing subtle details and outliers.

Displaying a large number of particles is similarly challenging in a scatter plot. This can be improved by drawing semi-transparent glyphs; however, frequency-based visualizations are better suited to visualize large datasets. Since approaches based on kernel density estimation or continuous scatter plots [BW08] are too expensive for our data sizes, we use histograms.

#### 4.2. Visualizing the Topology of Time-Dependent Flows

We now discuss the visualization of unsteady flows by selecting a sparse set of particles and their trajectories that best represent the flow topology. For particle-based flows, trajectories can be obtained by interpolating between the positions of particles over time. Note that these are not equivalent to massless tracer particles integrated in a velocity field. In particular, these particles are less affected by the exponential error growth near LCS. This is due to the particle-based simulations that apply attracting and repelling forces to the particles to enforce a uniform discretization of the domain. Moreover, these trajectories are well-suited to visualize particle-based fluid flows since they depict the actual movement of particles and do not depend on numerical integration. By brushing and linking, particles and trajectories can be selected intuitively using any of the views. Using the proposed per-particle LCS distance, trajectories near LCS are thus easily selected and visualized.

#### 4.3. Visualizing Mixing in Multiphase Fluid Flows

Multiphase fluid simulations contain particles of different types. By color mapping each particle according to its type, the distinct phases and their interface become visible, cf. Figure 6 (c). The visualization strongly indicates the presence of vortices and separatrices, i.e. separating lines with little cross flux, that are strongly related to the mixing behavior. To gain insight into the dynamics of the phase interface and the mixing regions over time, we identify the transport barriers in the form of LCS. As shown in Figure 1(b), we can display LCS together with the phase interface by emphasizing particles near a LCS. Although the FTLE also indicates the transport behavior,

cf. Figure 1(a), the sparse representation of the coherent structures significantly reduces visual clutter. Since the LCS are minimally diffusive, little to no mixing should occur across the boundaries. Accordingly, the separatrices coincide with the LCS, whilst vortices near phase interfaces indicate possible mixing regions. The LCS further show the behavior of the flow surrounding the interface, which influences the evolution of the interface over time.

### 5. Implementation

In this section, we outline a parallel implementation of the LCS computation and the spatial visualization of particle data.

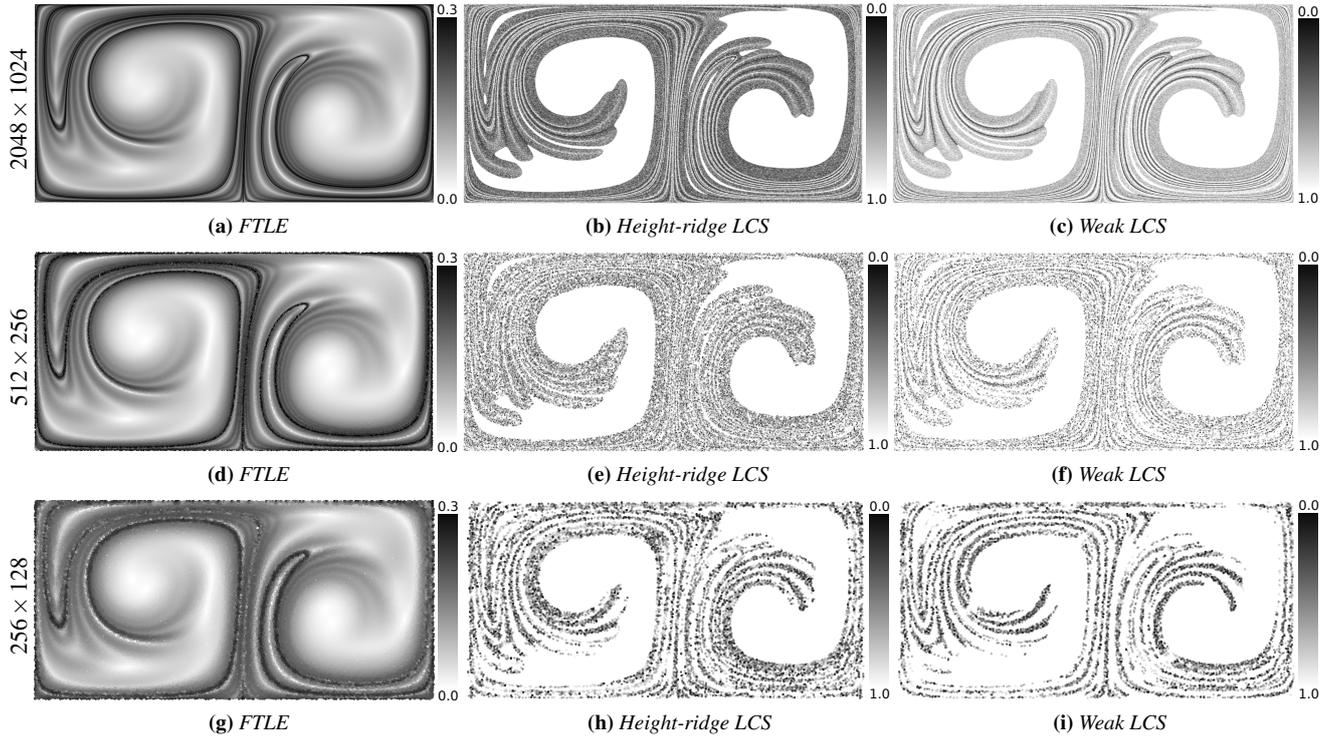
#### 5.1. Parallel LCS Computation

During the computation of the LCS, we need to repeatedly query the local neighborhood of a particle. In our implementation, we use a uniform grid to efficiently determine the neighborhood. If the cell sizes are chosen equal to the kernel support, a minimal number of cells have to be queried. As kernel function, we use a quintic spline [LL10].

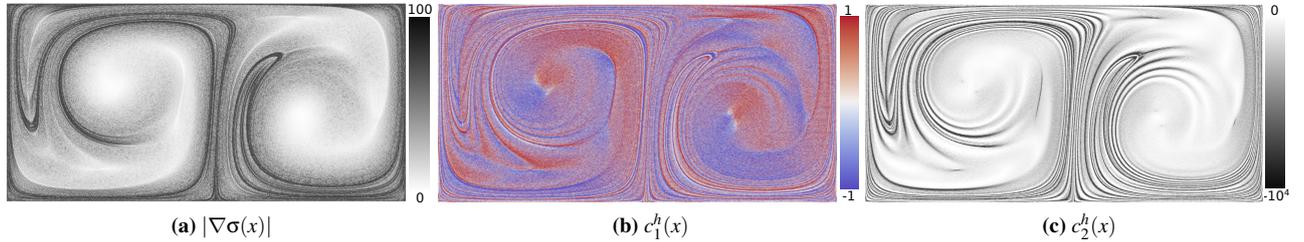
We perform all computations on the particles in parallel on the GPU by splitting them in two steps: First, we compute the spatial derivatives and evaluate the LCS criteria for every particle using its local neighborhood. In the second step, we compute the LCS distance for each particle. For a particle at position  $x$ , given either vector  $v_{\min}$  to detect height-ridges or  $\xi_n$  for the weak LCS definition, we determine the mean vector and the covariance matrix from the local neighborhood  $\mathcal{N}(x)$ . Then we extract the major eigenvector of the covariance matrix, according to which we orient the vectors of all particles in the neighborhood. Lastly, we evaluate the zero crossings and compute the distance from  $x$  to the closest LCS.

#### 5.2. Spatial Visualization

We depict particles as semi-transparent glyphs by applying a 1D transfer function to map a scalar value to color and opacity. Rendering particles as illuminated spheres gives a good perception of depth, but Gaussian kernels are better suited to illustrate complex, semi-transparent regions, see Figure 1. To render semi-transparent particles, back-to-front sorting lead to the best results. However, to render particle trajectories, sorting would be difficult. Instead, we use order-independent transparency [MKKP18]. Moreover, we take only a random subset when visualizing particle trajectories to reduce occlusion and visual clutter.



**Figure 4:** We evaluate our approach on the double gyre dataset with different data sizes. As a reference, the forward FTLE of the double gyre computed from the integrated particle trajectories is shown in the left column. The LCS distance mapped to color is shown in the middle using the height-ridge criteria and in the right column using the weak LCS criteria.



**Figure 5:** Visualization of the gradient magnitude (a), the first height-ridge criteria (b) that should be equal to zero, and the second criteria (c) that should be below zero at an LCS.

## 6. Experiments

To evaluate our approach, we apply it to a synthetic and a two- and three-dimensional SPH dataset.

### 6.1. Double Gyre

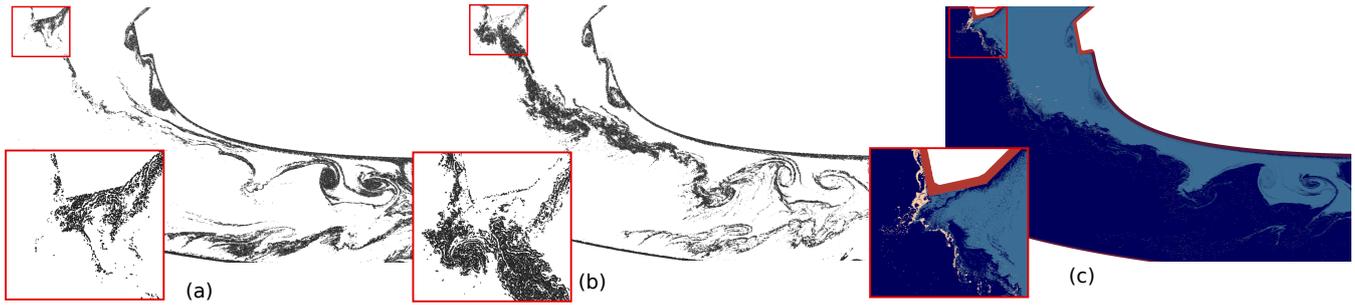
The double gyre is a two-dimensional periodic and unsteady vector field that describes two counter rotating gyres. It is commonly used for the validation of FTLE and LCS. We use the following definition:

$$v(x, y, t) = \begin{pmatrix} -\pi A \sin(f(x, t)\pi) \cos(\pi y) \\ \pi A \cos(f(x, t)\pi) \sin(\pi y) \frac{\partial}{\partial x} f(x, t) \end{pmatrix}, \quad (14)$$

where

$$\begin{aligned} f(x, t) &= a(t)x^2 + b(t)x \\ a(t) &= \varepsilon \sin(\omega t) \\ b(t) &= 1 - 2\varepsilon \sin(\omega t). \end{aligned} \quad (15)$$

We set  $A = 0.1$ ,  $\omega = 2\pi/10$ , and  $\varepsilon = 0.1$ . We convert this analytic vector field into a particle-based representation by integrating particles using a fourth order Runge-Kutta scheme and sampling the trajectories at fixed time steps of 0.1. The particles are uniformly seeded in the domain  $[0, 2] \times [0, 1]$ , but jittering is applied to reduce aliasing artifacts. In total, this leads to  $2048 \times 1024$  particles. Lastly, we compute the forward FTLE over the time interval  $[0, 20]$ . We apply an FTLE threshold of 0.08 to compute the LCS.



**Figure 6:** LCS from a large time interval (a) and a small interval (b). The fluid phases are shown in (c).

Figure 4 shows FTLE and LCS distance by applying a linear color map. The LCS distance visualizations in (b) and (c) depict the expected coherent structures, cf. e.g. Farazmand [FH12]. The ridges can also be visually inferred from the FTLE field in Figure 4(a). As expected, the height ridge and weak LCS criteria identify the same coherent structures, but the height ridge LCS are thicker.

The second and third rows of Figure 4 show the influence of the number of particles on the accuracy of the computation. Although the LCS distance still identifies the coherent structures even for the smallest dataset, it degrades faster than the FTLE. This is not surprising since the computation of the LCS distance requires the first and second derivatives of the FTLE.

Figure 5 shows a visualization of the gradient magnitude of the FTLE from the full dataset in (a) and the two height-ridge criteria in (b) and (c). These results emphasize that the derivatives and the derived quantities thereof are smooth, ensuring the robustness of the approach. Due to our approximation of the derivatives using kernel functions, increasing the size of the neighborhood leads to smoother derivatives, but will also smooth out finer features.

## 6.2. Fuel Spray

This two-dimensional dataset from a simulation of a fuel spray contains about 12.6 million particles per time step. As shown in Figure 6 (c), fuel particles (yellow) are injected on the top left and mix with two distinct gas phases (light and dark blue). The atomization of the fuel into a spray of fine particles in the surrounding gas phase is investigated by domain scientists to determine the quality and characteristics of the spray.

Figure 1 shows the backward FTLE (a) and the LCS together with the fluid phases (b). The LCS capture most of the ridges in the FTLE field and give a clear indication of the global transport and mixing behavior. On smaller scales, the LCS seem to lose some accuracy compared to the FTLE, indicating a lack of resolution. Besides enforcing an FTLE threshold of 0.1, we did not apply any smoothing or filtering operations, which would be required by grid-based extracting methods.

Since the computation of the FTLE and LCS is fast, it is possible to quickly explore different time intervals. As shown in Figure 6 (a) and (b), a large interval smooths out short-lived structures, whilst a small time interval emphasizes these, but might fail to detect longer-lived structures. We further correlate the LCS over different time intervals, for example using the parallel coordinate plot in Figure 3,

**Table 1:** Performance measurements of our datasets.

Dataset	Num. particles	FTLE	Height-ridge LCS	Weak LCS
Double Gyre	2,097,152	52.1 ms	10.2 ms	9.5 ms
Fuel Spray	12,629,195	370.4 ms	48.4 ms	48.2 ms
Bubble	4,347,225	102.7 ms	22.7 ms	20.6 ms

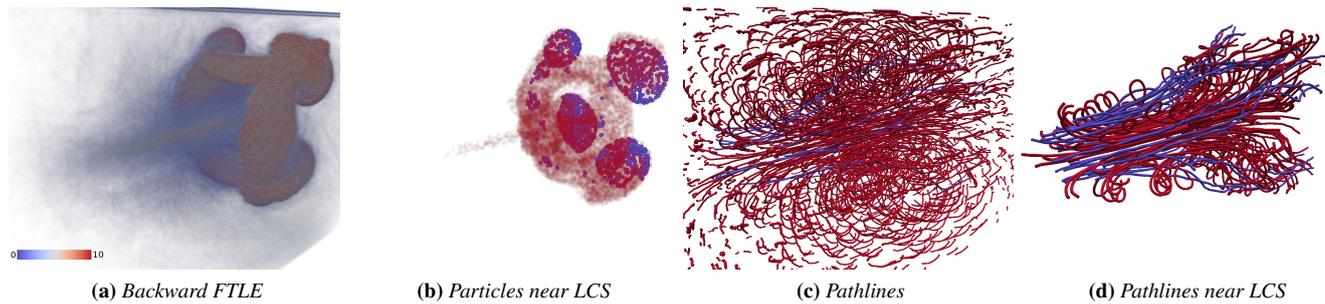
to determine which structures exist in one or both time scales. From the LCS in different time scales shown in Figure 6 (a) and (b) and the parallel coordinate plot, we gather that the vortical region (red) is short lived. The fluid phase visualization in (c) indicates strong mixing in this region caused by this vortex. However, the longer lived transport barriers in the middle of the complete flow show little cross flux and thus completely separate the phases. On the right, the transport barrier itself starts to swirl. Even though it still shows little cross flux, the phases start to mix on a greater scale.

## 6.3. Bubble

The Bubble dataset is a laminar, two-phase flow of an air bubble moving through water. The dataset can be reproduced with the GPUSPH [gpu] simulation code. The domain is 6 times the size of the spherical bubble and is discretized using 4.3 million particles in each of the 50 discrete time steps in the interval  $[0, 0.5]$ .

Figure 7 shows the backward FTLE (a) and particles close to the corresponding height-ridge LCS (b), which we color according to the fluid type. The LCS correspond to the ridges of the FTLE, but without the small-scale disturbances visible in the FTLE. The attracting coherent structures convey the transport of the air bubble that moves from left to right and starts to split. By visualizing both LCS and the fluid phases, the splitting of the air bubble in smaller bubbles becomes clearly visible.

In (c), we visualize particles trajectories, chosen by random from the whole dataset and colored according to their material. Although the different fluid phases become visible, the visualization suffers from significant amount of clutter due to a large number of short, and relatively uninteresting trajectories. By creating trajectories from particles near LCS, these trajectories are effectively filtered out in (d). The movement and splitting of the air bubble is better visible. In total, this reveals the major flow features with respect to the air particles and how they mix with the surrounding fluid.



**Figure 7:** Visualization of the Bubble dataset using the backward FTLE (a), particles near LCS (b), particle trajectories (c), and trajectories created from particles near LCS.

#### 6.4. Performance Measurements

Table 1 shows our performance measurements, which support our claim of an interactive visualization. Since the FTLE and LCS computations are extremely fast, different time scales can be efficiently explored. Moreover, the evaluation of height-ridge and weak LCS shows a similar complexity. All of our tests were run on an Intel i7 with 32GB of system memory and a Geforce 1080Ti graphics card with 11GB of dedicated GPU memory.

Since our visualization and filtering techniques are fast, our performance is bound by loading data from disk. Although we asynchronously load likely time steps in advance in a separate thread, requests for an unloaded time step might take a couple of seconds to complete for larger datasets. Operations that require loading multiple time steps, e.g. creating pathlines, are bottlenecked by the I/O operations.

#### 7. Discussion

As our experiments have shown, the proposed LCS distance is well-suited to identify the LCS and is fast to compute. Compared to the FTLE, the LCS distance is sparse and can thus be combined with other visualizations, such as coloring by fluid type for multiphase flows. Moreover, it allows us to select particles of interest, e.g. to visualize particle trajectories. However, since the computation of the LCS distance is based on first and second-order derivatives, it leads to a reduced accuracy compared to the FTLE and might thus miss features that are not sufficiently captured by the data.

Lastly, our comparison between the height-ridge and weak LCS definition is inconclusive. Although the weak LCS offers stronger theoretical guarantees, in our experiments we found that the most significant difference is the exclusion of LCS that stem from shearing in the flow.

#### 8. Conclusion

In this work, we have considered Lagrangian coherent structures for the interactive visualization of time-dependent particle flows. In detail, we formulate two different LCS definitions on the particle data that allows us to employ particle-based formulations for the spatial derivatives. We introduce a per-particle distance to the closest LCS to visualize and analyze particle-based flows. With multiple,

linked views on the data, we are able to effectively combine this distance with other particle quantities and features. In our experiments, we show the effectiveness of this approach to visualize Lagrangian dynamics and mixing behavior in multiphase fluids. Additionally, selecting and comparing different time scales can be effectively performed using the proposed visual analysis approach. Although this work has been an important step towards the interactive analysis of large particle-based flows, the steadily growing size of simulations remains a challenging issue.

#### References

- [AGJ11] AGRANOVSKY A., GARTH C., JOY K. I.: Extracting flow structures using sparse particles. In *Vision, Modeling, and Visualization* (2011), pp. 153–160. 2, 3
- [BGT12] BARAKAT S., GARTH C., TRICOCHÉ X.: Interactive computation and rendering of finite-time Lyapunov exponent fields. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (2012), 1368–1380. 2
- [BKKW08] BÜRGER K., KONDRATIEVA P., KRÜGER J., WESTERMANN R.: Importance-driven particle techniques for flow visualization. In *2008 IEEE Pacific Visualization Symposium* (2008), pp. 71–78. 3
- [BL99] BONET J., LOK T.-S.: Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering* 180, 1 (1999), 97–115. 3
- [BMDH07] BÜRGER R., MUIGG P., DOLEISCH H., HAUSER H.: Interactive cross-detector analysis of vortical flow data. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization* (2007), pp. 98–110. 2
- [BR10] BRUNTON S. L., ROWLEY C. W.: Fast computation of finite-time Lyapunov exponent fields for unsteady flows. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 20, 1 (2010), 017503. 2
- [Bro85] BROOKSHAW L.: A method of calculating radiative heat diffusion in particle simulations. *Publications of the Astronomical Society of Australia* 6, 2 (1985), 207–210. 3
- [BSK\*07] BÜRGER K., SCHNEIDER J., KONDRATIEVA P., KRÜGER J. H., WESTERMANN R.: Interactive visual exploration of unsteady 3d flows. In *EuroVis* (2007), pp. 251–258. 3
- [BW08] BACHTHALER S., WEISKOPF D.: Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1428–1435. 5
- [CFCN\*07] CUETO-FELGUEROSO L., COLOMINAS I., NOGUEIRA X., NAVARRINA F., CASTELEIRO M.: Finite volume solvers and moving least-squares approximations for the compressible Navier-Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics and Engineering* 196, 45 (2007), 4712–4736. 3

- [CRK12] CONTI C., ROSSINELLI D., KOUMOUTSAKOS P.: GPU and APU computations of finite time Lyapunov exponent fields. *Journal of Computational Physics* 231, 5 (2012), 2229–2244. 2
- [DRC\*18] DAUCH T., RAPP T., CHAUSSONNET G., BRAUN S., KELLER M., KADEN J., KOCH R., DACHSBACHER C., BAUER H.-J.: Highly efficient computation of finite-time Lyapunov exponents (FTLE) on GPUs based on three-dimensional SPH datasets. *Computers & Fluids* 175 (2018), 129–141. 2, 3
- [FH12] FARAZMAND M., HALLER G.: Computing Lagrangian coherent structures from their variational theory. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22, 1 (2012), 013128. 3, 7
- [FP01] FURST J. D., PIZER S. M.: Marching ridges. In *2001 IASTED International Conference on Signal and Image Processing* (2001). 4
- [GGTH07] GARTH C., GERHARDT F., TRICOCHÉ X., HANS H.: Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1464–1471. 2
- [GKM\*15] GROTTTEL S., KRONE M., MÜLLER C., REINA G., ERTL T.: Megamol – a prototyping framework for particle-based visualization. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (2015), 201–214. 3
- [gpu] GPUSPH. <http://www.gpusph.org>. Accessed: 2019-06-04. 7
- [GXY11] GUO H., XIAO H., YUAN X.: Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. In *2011 IEEE Pacific Visualization Symposium* (2011), pp. 19–26. 5
- [Hal01] HALLER G.: Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena* 149, 4 (2001), 248–277. 2
- [Hal11] HALLER G.: A variational theory of hyperbolic Lagrangian coherent structures. *Physica D: Nonlinear Phenomena* 240, 7 (2011), 574–598. 2, 3, 4
- [Hal15] HALLER G.: Lagrangian coherent structures. *Annual Review of Fluid Mechanics* 47 (2015), 137–162. 2
- [HW09] HEINRICH J., WEISKOPF D.: Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1531–1538. 5
- [HY00] HALLER G., YUAN G.: Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena* 147, 3 (2000), 352–370. 2
- [JMEL08] JONES C., MA K.-L., ETHIER S., LEE W.-L.: An integrated exploration approach to visualizing multivariate particle data. *Computing in Science & Engineering* 10, 4 (2008), 20–29. 3
- [KCH\*18] KINDLMANN G., CHIW C., HUYNH T., GYULASSY A., REPPY J., BREMER P.-T.: Rendering and extracting extremal features in 3D fields. *Computer Graphics Forum* 37, 3 (2018), 525–536. 4
- [KER\*14] KUHN A., ENGELKE W., RÖSSL C., HADWIGER M., THEISEL H.: Time line cell tracking for the approximation of Lagrangian coherent structures with subgrid accuracy. *Computer Graphics Forum* 33, 1 (2014), 222–234. 3
- [KKKW05] KRÜGER J., KIPFER P., KONCLRATIEVA P., WESTERMANN R.: A particle system for interactive visualization of 3D flows. *IEEE Transactions on visualization and computer graphics* 11, 6 (2005), 744–756. 3
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *ACM siggraph computer graphics* (1987), vol. 21, ACM, pp. 163–169. 4
- [LL10] LIU M. B., LIU G. R.: Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering* 17, 1 (2010), 25–76. 5
- [LS81] LANCASTER P., SALKAUSKAS K.: Surfaces generated by moving least squares methods. *Mathematics of computation* 37, 155 (1981), 141–158. 3
- [LZM\*11] LEŽ A., ZAJIC A., MATKOVIĆ K., POBITZER A., MAYER M., HAUSER H.: Interactive exploration and analysis of pathlines in flow data. *Proceedings of WSCG* (2011), 17–24. 2
- [MHP\*07] MATHUR M., HALLER G., PEACOCK T., RUPPERT-FELSOT J. E., SWINNEY H. L.: Uncovering the Lagrangian skeleton of turbulence. *Physical Review Letters* 98, 14 (2007), 144502. 2
- [MKKP18] MÜNSTERMANN C., KRUMPEN S., KLEIN R., PETERS C.: Moment-based order-independent transparency. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 7:1–7:20. 5
- [Mon92] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics* 30, 1 (1992), 543–574. 3
- [NH06] NOVOTNY M., HAUSER H.: Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 893–900. 5
- [RHD\*17] REINHARDT S., HUBER M., DUMITRESCU O., KRONE M., EBERHARDT B., WEISKOPF D.: Visual debugging of SPH simulations. In *21st International Conference Information Visualisation* (2017), pp. 117–126. 3
- [SCMZ16] SUN P., COLAGROSSI A., MARRONE S., ZHANG A.: Detection of Lagrangian coherent structures in the sph framework. *Computer Methods in Applied Mechanics and Engineering* 305, Supplement C (2016), 849–868. 2, 3
- [SGSM08] SALZBRUNN T., GARTH C., SCHEUERMANN G., MEYER J.: Pathline predicates and unsteady flow structures. *The Visual Computer* 24, 12 (2008), 1039–1051. 3
- [SLM05] SHADDEN S. C., LEKIEN F., MARSDEN J. E.: Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena* 212, 3 (2005), 271–304. 2
- [SP07] SADLO F., PEIKERT R.: Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1456–1463. 2, 4
- [SP09] SADLO F., PEIKERT R.: Visualizing Lagrangian coherent structures and comparison to vector field topology. *Topology-Based Methods in Visualization II* (2009), 15–29. 4
- [SPFT12] SCHINDLER B., PEIKERT R., FUCHS R., THEISEL H.: Ridge concepts for the visualization of Lagrangian coherent structures. *Topological Methods in Data Analysis and Visualization II* (2012), 221–235. 2
- [SRP11] SADLO F., RIGAZZI A., PEIKERT R.: Time-dependent visualization of Lagrangian coherent structures by grid advection. *Topological Methods in Data Analysis and Visualization* (2011), 151–165. 2
- [STH\*09] SHI K., THEISEL H., HAUSER H., WEINKAUF T., MATKOVIĆ K., HEGE H.-C., SEIDEL H.-P.: Path line attributes – an information visualization approach to analyzing the dynamic behavior of 3D time-dependent flow fields. In *Topology-Based Methods in Visualization II* (2009), pp. 75–88. 2, 3
- [STS10] SCHULTZ T., THEISEL H., SEIDEL H.: Crease surfaces: From theory to extraction and application to diffusion tensor MRI. *IEEE Transactions on Visualization and Computer Graphics* 16, 1 (2010), 109–119. 4
- [SZCC17] SHI L., ZHANG L., CAO W., CHEN G.: Analysis enhanced particle-based flow visualization. *Electronic Imaging* 2017, 1 (2017), 12–21. 2, 3