Technical report

Nr.: FIN-03-2013

Reflected Vector Fields for Finding FTLE Ridges

M. Schulze, C. Rössl, D. J. Lehmann, H. Theisel

Arbeitsgruppe Visual Computing

Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg

Nr.: FIN-03-2013

Reflected Vector Fields for Finding FTLE Ridges

M. Schulze, C. Rössl, D. J. Lehmann, H. Theisel

Arbeitsgruppe Visual Computing

Technical report (Internet)
Elektronische Zeitschriftenreihe
der Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg
ISSN 1869-5078

Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg

Impressum (§ 5 TMG)

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Der Dekan

Verantwortlich für diese Ausgabe:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Dirk J. Lehmann
Postfach 4120
39016 Magdeburg
E-Mail: dirk@isg.cs.uni-magdeburg.de

http://www.cs.uni-magdeburg.de/Technical_reports.html

Technical report (Internet)
ISSN 1869-5078

Redaktionsschluss: 12.10.2013

Bezug: Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Dekanat

Reflected Vector Fields for Finding FTLE Ridges

Maik Schulze, Christian Rössl, Dirk J. Lehmann, Holger Theisel

Abstract— The computation of FTLE fields and the ridges therein can be interpreted as a sampling problem: at discrete sample points, the flow map is computed by numerical integration. The quality of FTLE ridges strongly depends on the sampling density, i.e., on the number and distribution of samples for the flow map computation. For long integration times, FTLE ridges tend to become very sharp, and they may get close to each other. Thus, an extremely dense sampling is necessary to find the ridges. We introduce a novel approach to computing FTLE ridges that does not require high sampling densities. We define a modification of the original velocity field – called the reflected vector field – such that integration in this field gives particles converging to the ridges. This way, accuracy is not dominated by sampling density any more but instead by integration time. We compare the approach with sampling based approaches and show its usefulness especially for extremely long integration times and sharp ridges.

Index Terms—Flow Visualization, FTLE, LCS

1 Introduction

Lagrangian Coherent Structures (LCS) are prominent and promising approaches to extract and visualize the global behavior of time-dependent flow fields. Among the various alternatives that have been proposed in recent years, Finite Time Lyapunov Exponents (FTLE) are one of the most common representatives of LCS.

Ridges in FTLE fields are known to depict separating structures in the flow. Their reliable computation is challenging both in terms of accuracy and performance. This makes the computation of FTLE fields and ridges an interesting and active area of research, which has attracted researchers in Flow Visualization and other communities for some years.

Existing approaches for FTLE computation can be interpreted as a sampling problem. The flow map is computed on a number of discrete sample points. Its gradient is estimated at the samples, either by finite differences or by a suitable integration of the Jacobian of the velocity field. This way the accuracy of FTLE structures depends mainly on the density of the flow map samples. This motivated various approaches to adaptive sampling as well as to using temporal coherence for the samples in adjacent time steps.

There are two reasons why the sampling problem for FTLE computation is challenging. Firstly, increasing the number of samples is expensive, since every new sample requires the numerical integration of a path line. Secondly, FTLE ridges (i.e., the structures of main interest) tend to become very sharp and located close to each other with increasing integration time, see Kuhn et al. [12]. This means that for longer integration times the sampling density that is necessary to capture the FTLE ridges increases dramatically. Although FTLE originally was suggested for rather short integration times only, see Haller and Yuan [10], the application to very long integration times is desired because important properties are better fulfilled:

In fact, the longer the integration time is, the more the ridges are material separation structures, see Shadden et al. [27]. However, the strongly increasing sampling density required for increasingly long integration times is a fundamental limitation in computing FTLE ridges for extremely large integration times.

In this paper, we introduce a new approach to compute FTLE ridges that does not depend on the sampling density any more. For this, we introduce the concept of reflected vector fields and show that when integrating a suitable combination of original and reflected fields, particles converge to points on the ridge with increasing integration time. With this, we obtain FTLE ridges not by densely sampling the flow map but by integrating a low number of particles and connecting their end points. This way, the accuracy of the method is not dominated by the sampling density but by the integration time. This makes our techniques particularly useful for extremely long integration times and sharp and dense ridge structures.

The paper is organized as follows: Section 2 provides background and summarizes related work. Section 3 illustrates the main idea for a simple 1D example. Section 4 takes this idea to 2D and introduces the concept of reflected vector fields and shows basic properties. Based on this, we describe our algorithm in Section 5 and show results in Section 6. We discuss our approach in Section 7 and conclude with Section 8.

2 Background and related work

The observation of few particle trajectories provides a simple and intuitive tool for the analysis of time-dependent flows. A more abstract analysis that leads to a more compact representation of global flow behavior is the extraction of features such as flow transport barriers. These are curves in 2D or surfaces in 3D that are never intersected by any path line for given time interval, i.e., they separate regions of different flow behavior. A general definition of such flow features is given by Lagrangian Coherent Structures (LCS) [10, 7].

The probably most common approach to computing LCS is based on the finite-time Lyapunov Exponent (FTLE): Given is an unsteady vector field $\mathbf{v}(\mathbf{x},t)$. Then its *flow map* $\phi_t^\tau(\mathbf{x}) = \phi(\mathbf{x},t,\tau)$ maps a particle seeded at $(\mathbf{x},t)$ to its destination after a path line integration of $\mathbf{v}$ over a time interval $[t,t+\tau]$ (we assume $\tau > 0$). The (spatial) *flow map gradient* $\nabla\phi(\mathbf{x},t,\tau) = \frac{\partial}{\partial \mathbf{x}}\phi(\mathbf{x},t,\tau)$ encodes the separation of particles seeded near $(\mathbf{x},t)$, and FTLE is defined in terms of its largest singular value, i.e., the largest magnitude of separation. Commonly, FTLE is defined as the scalar value

$$\mathrm{FTLE}(\mathbf{x},t,\tau) = \frac{1}{\tau}\ln\sqrt{\lambda_{\max}(\nabla^T\nabla)}\,,$$

where $\lambda_{\max}$ denotes the maximum of the real, positive eigenvalues of the Cauchy-Green tensor $\nabla^T\nabla$ with $\nabla = \nabla\phi(\mathbf{x},t,\tau)$. A numerical computation of FTLE is straightforward for a finite-difference approximation of the flow map gradient, see Haller and Yuan [10, 5].

This is a very efficient and probably the most wide-spread method for computing FTLE. For a comparison of alternative methods see, e.g., Kuhn et al. [12].

The FTLE fields encode how much particles that are seeded in a small neighborhood separate after a path line integration. Therefore *ridges*[1] of these fields separate regions of different flow behavior: there are the structures that we are interested in. Most of the techniques used for FTLE ridge extraction come from ridge extractors in medical imaging. We mention local conditions obtained by relaxing conditions of extremal structures, see Eberly et al. [2], Lindeberg [15], and also Peikert and Sadlo [18]), topological/watershed approaches by Sahner et al. [24], second derivative ridges by Lipinski and Mohseni

[1] The general term *crease structures* includes valleys with similar properties. In this paper we only use the term ridge.

[16], or definitions based on extremal curvature structures, see Ohtake et al. [17]. The extraction of ridge surfaces in 3D fields is examined by Peikert and Sadlo [18] and Schultz et al. [26]. A discussion and comparison of ridge concepts for the visualization of LCS is given by Schindler et al. [25]. We particularly mention the approach by Kindlmann et al. [11], which obtains ridges by particles integration in the underlying scalar field. This can also be used for FTLE ridges. However, we note that this is different to our method, because the FTLE field has to be computed by sampling prior to ridge extraction.

FTLE ridges have been used for a variety of applications by Lekien et al. [14], Haller [6], Shadden et al. [28], and Weldon et al. [32]. Shadden et al. [27] show that ridges of FTLE are approximate material structures, i.e., they converge to material structures for increasing integration times. This fact was used by Sadlo and Weiskopf in [23] to extract topology-like structures to accelerate the FTLE computation in 2D flows. A similar objective is pursued by Lipinski and Mohseni in [16] with a ridge tracking that is based on particle tracking. Since FTLE ridges are approximate but not perfect material structures, this induces a small error which makes the approach not applicable for an exact tracking of sharp ridges. The smallest FTLE values are additionally explored by Haller and Sapsis in [9]. Furthermore, different approaches to increasing performance, accuracy, and usefulness of FTLE as a visualization and visual analytics tool have been proposed [21, 4, 3, 20, 22, 13, 1]. Haller and Beron-Vera [8] show that transport barriers are shadowed by certain minimal geodesics. The barriers are obtained by a filtering of strainlines and shearlines and advection of the filtered segments. Strainlines and shearlines are in turn computed as the solutions of ordinary differential equations.

Ridges can become extremely sharp for long integration times. This fact is demonstrated in a recent benchmark on accuracy of FTLE computation by Kuhn et al. [12].

The results in this work reveal the limits of standard ridge extraction, e.g., from image analysis, for FTLE ridges in practice: the required sampling resolutions become extremely high. On the other hand, for long integration times there is less flux across FTLE ridges, which makes them a better approximation of LCS, see Shadden et al. [27]. Thus, FTLE for long integration times and sharp ridges are interesting in practice.

Üffinger et al. [30] generalize the work by Sadlo and Weiskopf [23] by finding separating structures based on an intersection of ridges in forward and backward FTLE. These structures are used (with a certain offset) as seed structures for generalized streak surfaces. Although the integration from the separating structures is similar to our approach, the construction of these structures differs fundamentally from our approach. In particular, [23] still rely on a high sampling for finding the separating structures.

3 A Simple Example

We explain the main idea of our approach by means of a very simple example: the 1D vector field

$$\mathbf{v}(x,t) = x\,(1-x)\,(1+x)\,,$$

which is of such simple structure that a closed form solution for the flow map (and similarly for FTLE) exists:

$$\phi_t^\tau(x) = \frac{x}{\sqrt{x^2 + e^{-2\tau}\,(1-x^2)}} \tag{1}$$

The ridge of forward FTLE is just the location $x = 0$. Although the ridge is known, we emulate a standard ridge extraction technique for an unknown ridge in order to estimate its accuracy. We restrict ourselves to the domain $x \in [-1,1]$. In order to get the ridge, we sample the domain and compute the flow map from the sample points. This gives information about the location of the ridge at an accuracy of the sampling density. We assume an adaptive sampling by binary partition of sampling intervals. Then for a desired accuracy of $\varepsilon > 0$ for the location of the ridge, we need at least $\log_2 \frac{1}{\varepsilon}$ flow map samples. Figure 1 illustrates this for $\varepsilon = \frac{1}{4}$. Although the number of samples

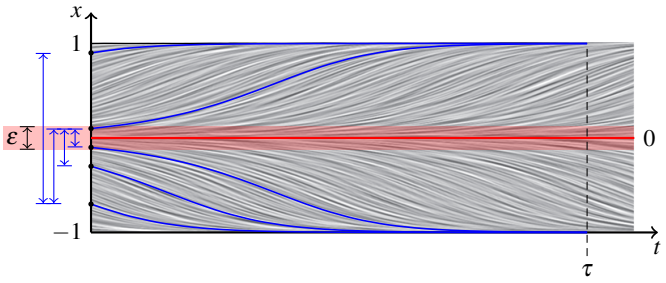

Fig. 1. Traditional adaptive ridge sampling by recursive bisection. Starting from an interval size 1.6, five samples are required to ensure an error bound of $\varepsilon = \frac{1}{4}$ (bisection intervals shown left).

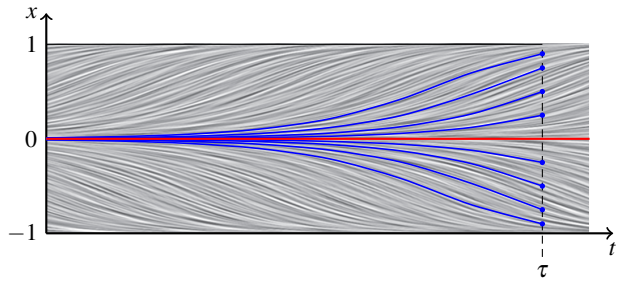

Fig. 2. Ridge sampling by backward integration. The sample points tend to converge quickly towards the ridge.

increases only a logarithmically with linearly increasing accuracy, this is a serious limitation because in practice ridges are not computed in 1D but in 2D or 3D domains.

In our simple example, the alternative to sampling-based forward FTLE computation is backward integration. Instead of sampling the domain at the time $t = 0$, we apply a backward integration from the time τ to 0. In our example we use eight equidistant samples. Figure 2 illustrates this. In this example we integrate backwards, i.e., we compute $\phi_\tau^{-\tau}(x)$ for $x \in \{\pm\frac{9}{10}, \pm\frac{3}{4}, \pm\frac{1}{2}, \pm\frac{1}{4}\}$. With this, we compute the ridge location together with an estimation of its accuracy by backward integrating a few path lines. Doing so, accuracy is dominated not by the sampling density but by integration time. We see that all lines end at time $t = 0$ very close to the ridge. (In fact, this holds for any line except those seeded at $x = \pm 1$.) With (1) given, it is easy to see that the distance between backward integrated particles and the ridge decreases exponentially in τ. This is independent of the starting point. This means that for *long* integration times τ, we get extremely accurate ridge locations just by backward integration of few particles. Moreover, the difference between two end points at $t = 0$ is a measure of how accurate the location of the found ridge is.

Unfortunately, this simple approach of forward FTLE ridge extraction by backward integration does not directly carry over from 1D to 2D vector fields. The reason is that for 1D vector fields, backward integration of a diverging area yields a converging area, i.e., an area where path lines converge to each other under backward integration. In 2D flows, we are mainly interested in LCS occurring in hyperbolic regions. Unfortunately, the backward integration of a hyperbolic region gives a hyperbolic region as well. This means that a simple backward integration does generally not converge. In the following we introduce the concept of reflected fields that provides the extension of the illustrated idea to 2D flows: these reflected vector fields are constructed such that backward integration in a hyperbolic region gives an attracting region.

4 Reflected Vector Fields

There can be several flow phenomena causing LCS. Here we are interested in hyperbolic regions, which can be interpreted as saddle points

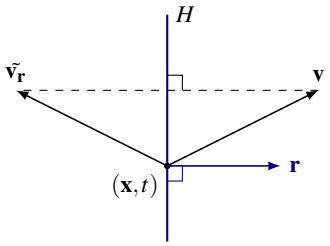

Fig. 3. $\tilde{\mathbf{v}}_\mathbf{r}$ is obtained by reflecting $\mathbf{v}$ about a line H, which is defined by its normal vector $\mathbf{r}$. The reflecting field $\mathbf{r}$ is determined locally at $(\mathbf{x},t)$ from the eigenvectors of the Jacobian.

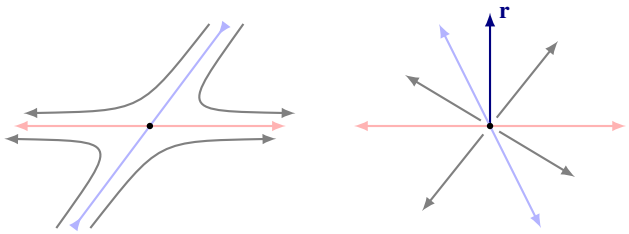

Fig. 4. Reflection near saddle with $\lambda_1 < 0 < \lambda_2$. The reflection of the original vector field $\mathbf{v}$ (left) about the axis spanned by the eigenvector $\mathbf{e}_2 \perp \mathbf{r}$ yields a source for the reflected field $\tilde{\mathbf{v}}_\mathbf{r}$ (right).

moving over time. Due to the unsteadiness, the separation does not occur at the saddle point but at a point in its neighborhood. The location of this point generally cannot be computed by a local analysis of $\mathbf{v}$. In order to find this point, we introduce a modification $\mathbf{w}$ of $\mathbf{v}$ such that its backward integration stably converges to the separating point.

The main idea is to reflect the vectors of 2D unsteady vector field $\mathbf{v}(\mathbf{x},t)$ about a certain line or plane. We assume a local coordinate system with origin $\mathbf{x}$. Any reflection plane H that contains the origin can be characterized by a normal vector $\mathbf{r} \neq \mathbf{0}$ as $H = \{\mathbf{y} \,|\, \mathbf{y}^\mathrm{T}\mathbf{r} = 0\}$. A reflection about H can be written as a linear operator

$$\mathbf{H}_\mathbf{r} \;=\; \mathbf{I} - 2\,\frac{\mathbf{r}\,\mathbf{r}^\mathrm{T}}{\mathbf{r}^\mathrm{T}\mathbf{r}}\;, \tag{2}$$

where $\mathbf{I}$ is the identity. $\mathbf{H}_\mathbf{r}$ is the well-known Householder reflector, in our case a matrix $\mathbf{H}_\mathbf{r} \in \mathbb{R}^{2\times 2}$. With the local reflections being characterized by a vector, we can define the *normal field* $\mathbf{r}(\mathbf{x},t)$ as a vector field. (We omit the space-time location $(\mathbf{x},t)$ in the following and write $\mathbf{r}$, $\mathbf{v}$, etc. for short.) Assume $\mathbf{r}$ is given, then we can define:

Definition 1 *Given is a vector field* $\mathbf{v}$ *and a non-vanishing normal vector field* $\mathbf{r}$. *The* reflected vector field $\tilde{\mathbf{v}}_\mathbf{r}$ *of* $\mathbf{v}$ *with the normal field* $\mathbf{r}$ *is defined as*

$$\tilde{\mathbf{v}}_\mathbf{r} = \mathbf{H}_\mathbf{r}\,\mathbf{v}\;. \tag{3}$$

Figure 3 gives an illustration. Reflected vector fields have the following properties:

- reproduction of critical points: $\tilde{\mathbf{v}}_\mathbf{r} = \mathbf{0} \Leftrightarrow \mathbf{v} = \mathbf{0}$.
- scale independence of $\mathbf{r}$: $\tilde{\mathbf{v}}_\mathbf{r} = \tilde{\mathbf{v}}_{s\mathbf{r}}$ for any $s \neq 0$.

Both properties can be easily verified from the definition of $\mathbf{H}_\mathbf{r}$. Note that scale independence includes orientation independence: $\tilde{\mathbf{v}}_\mathbf{r}$ does not depend on the orientation of $\mathbf{r}$. This qualifies eigenvector fields to act as normal fields.

Since we are interested in transforming a hyperbolic region of $\mathbf{v}$ into a parabolic region in $\tilde{\mathbf{v}}_\mathbf{r}$, we study the Jacobian of $\tilde{\mathbf{v}}_\mathbf{r}$. Let $\mathbf{J}$ be the Jacobian of $\mathbf{v}$. Then we get the Jacobian $\tilde{\mathbf{J}}_\mathbf{r}$ of $\tilde{\mathbf{v}}_\mathbf{r}$ by differentiating both sides of (3). From this we get the following property:

$$\mathbf{v} = \mathbf{0} \;\Rightarrow\; \det\mathbf{J} = -\det\tilde{\mathbf{J}}_\mathbf{r}. \tag{4}$$

The proof of (4) is a straightforward computation. It means that for 2D vector fields, a hyperbolic critical point (i.e, a saddle) is transformed to a parabolic critical points (i.e, a source, sink, or center) in $\tilde{\mathbf{v}}_\mathbf{r}$. Figure 4 gives an illustration of a saddle turning into a source after reflection: this is what we intend. Figure 5 shows the inflow and outflow near a saddle for different reflection axes.

Our goal is to find a normal field $\mathbf{r}$ such that the reflected field of a saddle is a source node. For this, we formulate $\mathbf{r}$ in terms of the eigenvectors of $\mathbf{J}$: let λ_1, λ_2 be the eigenvalues of $\mathbf{J}$, we assume that they are real and $\lambda_1 \leq \lambda_2$. Furthermore, let $\mathbf{e}_1, \mathbf{e}_2$ be the corresponding eigenvectors in $\mathbf{J}$, and let $\mathbf{g}_1, \mathbf{g}_2$ be the corresponding eigenvectors in $\mathbf{J}^\mathrm{T}$. ($\mathbf{J}$ and $\mathbf{J}^\mathrm{T}$ have the same eigenvalues but generally different eigenvectors). Then we define $\mathbf{r}$ to be perpendicular to $\mathbf{e}_2$ by setting $\mathbf{r} = \mathbf{g}_1$. We

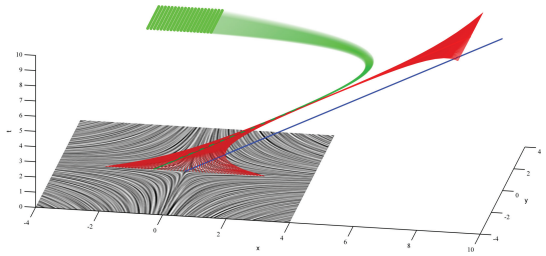

Fig. 6. Linearly moving saddle: while path lines of $\mathbf{v}$ show a hyperbolic behavior (i.e., diverge from each other in both forward and backward direction), the backward integration of $\mathbf{w}$ (green lines) converges stably towards the moving separation point.

are mainly interested in LCS occurring in hyperbolic regions, where no rotation is present. In such regions we have $\lambda_1 \leq 0 \leq \lambda_2$. For $\mathbf{v} = \mathbf{0}$, $\tilde{\mathbf{J}}_{\mathbf{g}_1}$ has the eigenvalues $0 \leq -\lambda_1, \lambda_2$, describing a parabolic diverging region. This means that a backward integration of $\tilde{\mathbf{v}}_{\mathbf{g}_1}$ gives a converging behavior of particles. Figure 4 illustrates this.

Up to now, our concept of vector field reflection transforms a steady saddle into steady source. For unsteady flows, we have to incorporate the moving of the saddle over time. For this, we consider the field

$$\mathbf{f} = \frac{1}{\det(\mathbf{v}_x, \mathbf{v}_y)} \begin{pmatrix} \det(\mathbf{v}_y, \mathbf{v}_t) \\ \det(\mathbf{v}_t, \mathbf{v}_x) \end{pmatrix}, \tag{5}$$

which corresponds to the feature flow field to track critical points over time in unsteady 2D vector fields [29]. We have to remove the saddle movement before applying the reflection: we define the time-dependent reflected vector field $\mathbf{w}$ as

$$\mathbf{w} = \mathbf{H}_{\mathbf{g}_1}\mathbf{h} + \mathbf{f}. \tag{6}$$

with $\mathbf{h} = \mathbf{v} - \mathbf{f}$. Note that $\mathbf{w}$ is only defined in regions where $\mathbf{v}$ is hyperbolic, i.e., where $\det(\mathbf{v}_x, \mathbf{v}_y) < 0$. Also note that $\mathbf{w}$ is computed from $\mathbf{v}$ only by local operations. In particular, we need $\mathbf{v}$ and its Jacobian only to compute $\mathbf{w}$ locally.

We illustrate the concept of time-dependent reflected vector fields for a simple example: the field

$$\mathbf{v}(x,y,t) = \begin{pmatrix} -(x - t\,u_d) \\ y - t\,v_d \end{pmatrix}, \tag{7}$$

which describes a saddle moving with constant speed $(u_d, v_d)^T$ over time. Figure 6 illustrates this for $(u_d, v_d)^T = (1,0)^T$. The LIC plane shows $\mathbf{v}$ at $t = 0$, and the blue line $(t\,u_d, t\,v_d)$ denotes the path of the saddle over time. The moving separating point is located along the line $(-u_d + t\,u_d, v_d + t\,v_d)^T$, which is in this case the only straight path line of $\mathbf{v}$. It is shown in the figure by integrating the red curves in $\mathbf{v}$. Note that this line has a hyperbolic behavior: the red lines diverge from it both in forward and in backward direction. The alternative to computing the moving separating point is to backward integrate $\mathbf{w}$. Here, the seed points are the green points in the upper left (which are

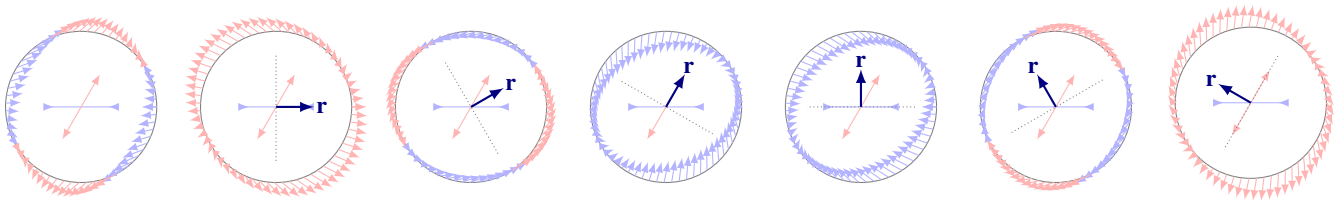

Fig. 5. Inflow and outflow near a saddle in the vector field $\mathbf{v}$ (left) and reflected fields $\bar{\mathbf{v}}_\mathbf{r}$ for different constant reflections $\mathbf{r}$. The reflection vector $\mathbf{r}$ and the associated reflection axis (dotted) are turned counter clockwise. (All figures show the original directions of eigenvectors.)

actually rather far away from the line). The backward integration of $\mathbf{w}$ (green lines) converges stably to the desired separation line in space-time.

5 Algorithm

Based on the idea of reflected vector fields, we propose the following algorithms for the extraction of FTLE ridges. The core contribution is the first step of constructing seed structures. The subsequent ridge tracking starting from a seed is essentially a 1D curve sampling with an adaptive parametrization, similar to a 1D integration.

5.1 Generation of seed curves

Given is the vector field $\mathbf{v}(\mathbf{x},t)$ over a spatial domain D and the integration time τ. We pick a discrete set of particle samples in D at time τ and perform a backward integration of the time-dependent reflected field $\mathbf{w}$ as defined in (6). As we are interested in hyperbolic regions, we seed particles only in these regions at time τ. Furthermore, we stop integration for particles that would leave hyperbolic regions, i.e., some particles may not reach the destination at $t = 0$. The trajectories of the particles converge rapidly to curves that end on ridges. In fact, the behavior is similar to the simple 1D example (see Figure 2) as shown for instance in Figures 6, 9 (top left) or 11 (bottom). For this reason, there is no need for a dense initial particle set. In particular, particle density does not correlate with "sharpness" or distance of ridges! This is demonstrated in Figure 11 (bottom). We remark that in practice more samples increase the chance tracking the ridge accurately over a long time (see below). However, the number of samples is typically orders of magnitude lower than for traditional ridge computation.

5.2 Seed curve selection

Backward integration of $\mathbf{w}$ from samples gives a potentially large set of candidate seed curves, which end near a ridge for $t = 0$. As the curves converge to trajectories of separation points, their end points form tight clusters in D (see, e.g., Figure 9). We perform an outlier filtering by discarding the 30% of points with largest norm $\|\mathbf{J}\|_F$. We then use a standard mean-shift clustering algorithm to identify clusters within the remaining points. Then small clusters consisting of only few points are discarded. From each remaining cluster we pick the one with the shortest particle trajectory: this trajectory is selected as seed curve for this cluster. This process is motivated by the fact that points that have not converged as well as other points during integration exhibit stronger saddle-like behavior in their local neighborhood. The outlier rejection step significantly eases the clustering step as it increases the ratio of inter-cluster to intra-cluster distance. Figure 7 illustrates this setup after backward integration in the reflected field: black points are discarded, blue and yellow points are then clustered. End points of selected seed curves are depicted green.

Note that the clusters and also the trajectory bundles are very tight (e.g., a cluster radius in the order of 10^{-5} for Figure 9). Still, we aim at getting as close as possible to the true separation point to ensure accurate ridge tracking. Our experiments show that the above filtering and selection heuristic tend to find very good seed curves.

5.3 Ridge tracking

Each seed curve represents an approximation to the trajectory of a separating point, and we use these curves for ridge tracking. The seed

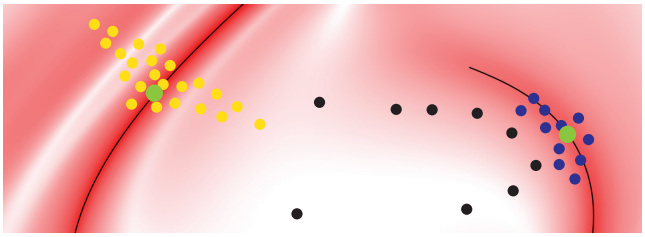

Fig. 7. Illustration of seed curve selection setup after backward integration in the time-dependent reflected field $\mathbf{w}$. Outliers (black) are removed and clusters (yellow and blue) are computed. For each cluster a candidate is selected by minimal arc length of integral curve in $\mathbf{w}$ (green).

curves are given as spline curves $\mathbf{s}(t)$ from the numerical integrator. (In our implementation, we a fourth order Runge-Kutta scheme and represent trajectories as cubic Hermite splines.) The basic idea of the ridge tracking is simple: Starting from $t = 0$ we evaluate the seed curve and determine the vector field Jacobian $\mathbf{J}(\mathbf{s}(t),t)$ for increasing t. As we are in a hyperbolic region we obtain real eigenvalues $\lambda_1 < 0 < \lambda_2$ and associated eigenvectors $\mathbf{e}_{1,2}$. From each curve point we take a small offset in direction of $\mathbf{e}_2$ to pick an offset point $\mathbf{s}' := \mathbf{s}(t) + \gamma\mathbf{e}_2$. Backward integration of the vector field $\mathbf{v}$ starting from $(\mathbf{s}',t)$ yields a point on the ridge at $t = 0$. This way, we can sample the ridge in an interval $[0, t_{\max}]$, where the limit $t_{\max} \leq \tau$ must ensure that $s(t)$ is still close to the trajectory of the separating point. In practice, we limit the deviation from other seed curves in the same cluster/trajectory bundle. Note that a consistent orientation of the eigenvector $\mathbf{e}_2$ is required along $s(t)$. This can be enforced easily, e.g., by taking the previous/nearest evaluated $\mathbf{e}_2$ as reference.

The parameter $\gamma \neq 0$ is a small signed number. Its sign determines, which direction of the ridge is followed. Its magnitude trades accuracy ($|\gamma|$ small) versus total length of the extracted ridge ($|\gamma|$ large). For our experiments we use $|\gamma| \in \left[d \cdot 10^{-2}, d \cdot 10^{-3}\right]$, where d measures the extent of the domain as length of the bounding box diagonal. The magnitude of γ is the only "critical" parameter of our approach in a sense that firstly, it should be selected depending on the input data $\mathbf{v}$, and secondly, for a "wrong" choice ridges become inaccurate or are partially lost. We currently cannot give a reliable automatic and or even time-dependent selection (see also discussion in Section 7).

Based on the above, we can now assume the ridge is given as a curve $\mathbf{c}$, where each curve point $\mathbf{c}(t)$ is evaluated as described above. A naïve – e.g., uniform in t – sampling of $\mathbf{c}(t)$ will not yield satisfactory results in practice because the parametrizations of $\mathbf{s}$ and $\mathbf{c}$ differ significantly. This means that the curves differ significantly in speed, or more formally the ratio

$$\| \tfrac{d}{dt}\mathbf{s}(t) \| \,/\, \| \tfrac{d}{dt}\mathbf{c}(t) \|$$

generally ranges from extremely large to extremely small with considerable variation within the time interval of interest. This is not surprising as the evaluation of the ridge is expected to show an exponential behavior. We propose a simple and effective solution for this 1D sampling problem. Ultimately, we are interested in the arc length of $\mathbf{c}$,

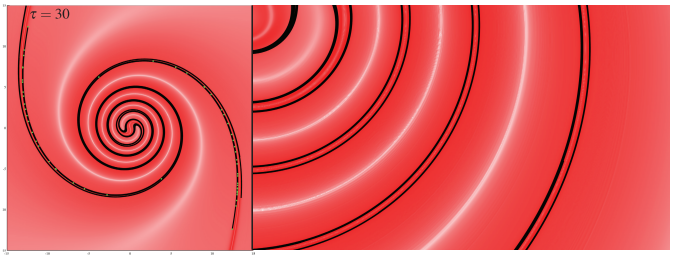

Fig. 8. Ridges of spiral focus with parameter $p_0 = 16$.

i.e., for a prescribed distance Δs measured in D we determine the time difference Δt such that

$$\| \mathbf{c}(t+\Delta t) - \mathbf{c}(t) \| \approx \Delta s \; .$$

We achieve this by growing Δt followed by a recursive bisection in Δt until the above condition is satisfied within some error bound. (We ensure a maximum relative error of $\Delta s/100$.) This way we can sample the ridge as a polyline with approximately uniform segments of length Δs.

Based on this we adapt the local segment length Δs to the shape of the ridge. Our goal is to match the "true" speed of the ridge with the approximate speed $\Delta s/\Delta t$. We compute a local segment length Δ that satisfies this property approximately as follows: For each ridge segment we start with $\Delta = \Delta s$. We estimate $\frac{d}{dt}\mathbf{c}(t)$ using a finite-difference for a small time interval ($\delta t = 10^{-9}$). Then we adapt (grow or shrink) Δ iteratively until the ratio $\rho = \| \frac{d}{dt}\mathbf{c}(t) \| / \Delta$ is bounded such that $\max\{\rho, 1/\rho\} < \frac{3}{2}$. In addition, we bound the segment length and stop adaptation whenever $\Delta < \Delta s/10$ or $\Delta > \frac{5}{4}\Delta s$. The given error bounds and thresholds served our needs for all examples, and their choice is uncritical. We do not see requirement for second order adaptation incorporating curvature or the ridge.

6 Results

We apply our method to synthetic and measured data sets with different properties. We visualize extracted ridges as black lines that are superimposed on a forward-FTLE texture using the color map proposed by Garth et al. [4].

For all our experiments we use an adaptive fourth-order Runge-Kutta integrator. The analytic flow fields and their Jacobian are evaluated "exactly" from formulas, sampled data is reconstructed by tricubic C^1 interpolation.

Spiral Focus

We apply our method to the *spiral focus* vector field proposed as an FTLE benchmark by Kuhn et al. [12]. It is characterized by a focus sink in the shape of Fermat's spiral. The field parameter p_0 steers sharpness and closeness of ridges. We use $p_0 = 16$ and an integration time $\tau = 30$, this is a nontrivial choice (see [12]). Figure 8 shows the ridges extracted by our method. We obtain two ridge line pairs spiraling inward and meeting at the center. While the underlying FTLE texture shows the general separation behavior and shows consistence with the ridge lines, it is difficult to see the small valley line between pairs of ridges. The closeup on the right shows the bottom right corner of the data set. Bright valley lines are visible between adjacent ridges. The green dots show the cluster representative used for backward pathline integration. Note that we do not check for different representatives on the same ridge. Indeed, here our algorithm extracts ridge lines multiple times such that the visualization shows lines that are on top of each other. The fact that this is hardly visible indicates high accuracy of ridge detection independent of the particular seed.

Double Gyre

We also apply our method to the well-known double gyre vector field that was symmetrically extended such that we have the separating point away from the boundary. The primary feature within the flow is a saddle that moves periodically on the x-axis near the center $(1,0)$ creating a sinusoidal trajectory in space-time. We show our results in Figure 11 for different integration times $\tau = \{10, 15, 20, 25, 30, 35, 40\}$, i.e., for one to four periods. We analyze the double gyre in two compartments $[0,2] \times [-1,1]$ and can observe the black ridge line emanating from the saddle near the center. Due to the space-filling nature of ridges in this flow we show a close-up of the extracted ridges for the largest integration times. The top left image shows the space-time trajectories of backward-integrated reflection lines and their seed points. This gives a good impression of cluster tightness.

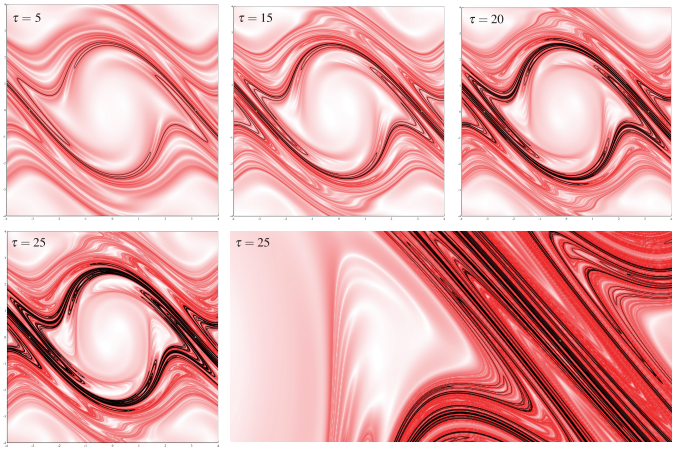

Fig. 10. Perturbed pendulum is a chaotic dynamical system. This phase portrait exhibits complex FTLE ridges. Our method stably extracts four black ridges for long integration times $\tau = \{5, 15, 20, 25\}$.

Perturbed Pendulum

Figure 10 shows the phase plane of a perturbed pendulum. The chaotic behavior of this dynamical system becomes visible with FTLE. We show our results for different integration times $\tau = \{5, 15, 20, 25\}$. Reflection lines integrated from the grid with dimension 20×20 converge rapidly to the source in $\mathbf{w}$ at $t = 0$ (shown as green dots). Backward pathline integration off the two curves results in four black ridge curves. Our method extracts the complex ridge geometry stably for large integration times such as $\tau = 25$. A close-up of the ridges near the right reflection field source is shown in the bottom right image.

PIV Cylinder

Figure 11 shows our results for the flow around a cylinder. The flow was measured using Particle Image Velocimetry and is defined on a regular grid with the dimension $560 \times 160 \times 61$ and a time span of $t \in [0, 480]$. The flow is fully developed at initial time $t = 0$ and contains the Kárman vortex street phenomenon. Vortices of opposite rotation appear behind the circular obstacle.

The top left image shows ridges extracted by our method for $\tau = 150$ in black. Reflection lines are seeded on a 300×100 grid and converge to multiple reflection field sources during backward integration. The clusters of the converged points are depicted by yellow dots with selected cluster representatives of green color. The integrated ridges become shorter with growing distance from the cylinder. This effect is consistent with the FTLE field, which becomes increasingly smooth in these regions.

The bottom left image shows the space-time trajectories of reflection lines of each cluster in distinct colors. A much coarser grid of dimension 50×16 is used for the bottom right figure. Despite a highly reduced number of seed points in the reflection field our method is able to extract smooth ridges.

The top right image in Figure 11 shows ridges for an increased integration time of $\tau = 250$ from a dense seed grid. Smaller clusters size

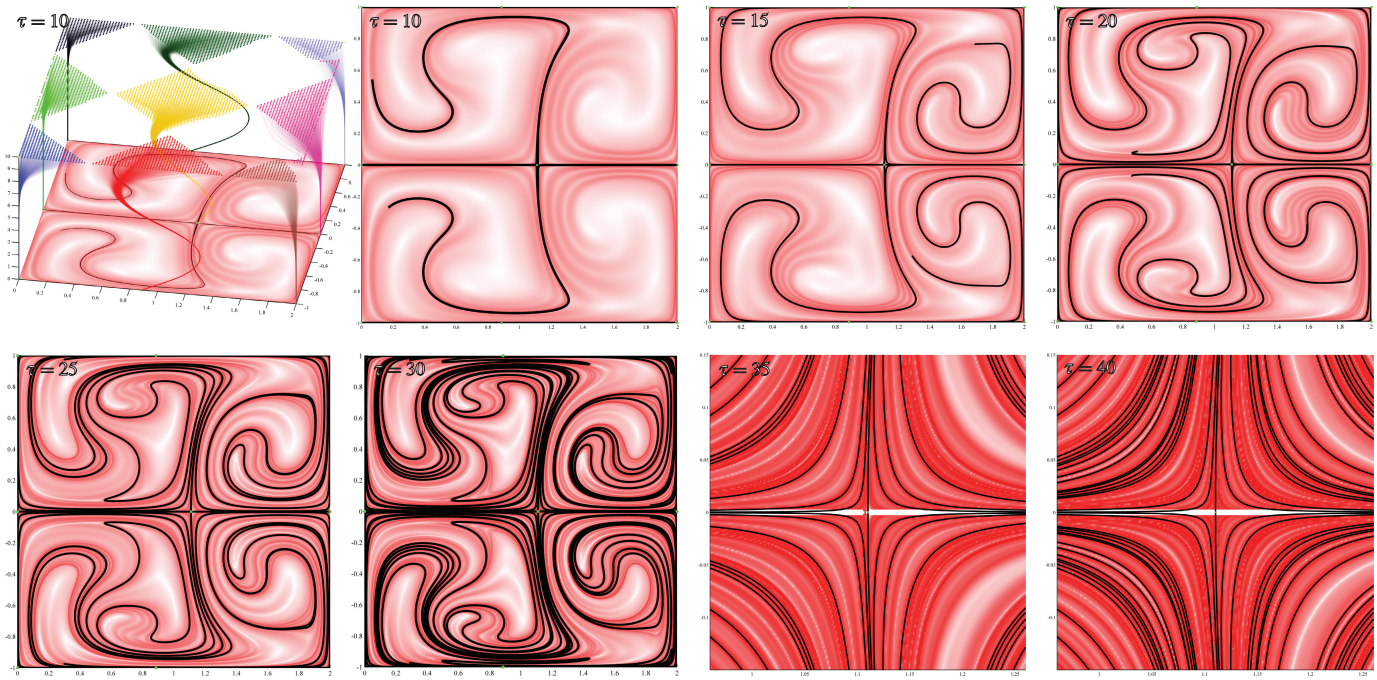

Fig. 9. Ridges of double gyre for different integration times τ. The top left image shows space-time trajectories in the reflected field. For $\tau = 35$ and $\tau = 40$ closeups on the separating saddle point can been.

than in the experiment to the left with $\tau = 150$ indicate an increasing precision of backward integration in the reflection field.

Performance

The performance sampling-based FTLE ridge extraction methods correlates with the smoothness of the results. This is not the case for our method. However, in order to obtain reliable results it is necessary to provide a reasonable seed grid for particle advection in the reflected field. Additionally, the obtained separating point trajectories are seed curves for pathline integration to obtain ridge curves. The quality of the ridge curves depends on the sampling density during this phase, leading to a tradeoff between quality and computation time. Significant portions of our algorithm can be run in parallel: the initial particle advection based on numerical integration is easily parallelized. We observe that filtering and clustering requires only milliseconds on in all experiments. Finally, ridge sampling off cluster candidates can be run in parallel as well.

We provide timings of these phases for different data sets that have been discussed in this section in Table 1. The last three columns show timings for the different stages of the algorithm: integration in $\mathbf{w}$ for seed generation, the selection of the seed curves (including clustering), and tracking (i.e., mainly integration in $\mathbf{v}$). The measurements were taken on a desktop computer with an Intel i7-2600K processor. The timings reveal that the run-time of our algorithm is governed by the numerical integration. Note that the cost for tracking depends on the prescribed target segment length Δs, which is chosen rather small (please zoom into the pictures).

Convergence and numerical stability

The cylinder data set is a challenge for any ridge extraction method: it exhibits a complex flow that is given as discrete samples. We applied our method to the cylinder data (see Figure 11) and obtain smooth ridges that are consistent with FTLE textures. (We do not have a better ground truth.) Repeating the experiment for a low resolution sampling reveals that the smoothness of ridges (and mostly also their length and location) does not correlate with the resolution of the particle seed grid. This is an advantage over existing methods such as subdivision-based image techniques for ridge tracking.

data set	τ	grid	integration	selection	tracking
cylinder	150	300×100	269.4s	0.11s	135.5s
cylinder	150	50×16	6.5s	0.1ms	121.6s
double gyre	10	100×100	2.5s	0.2ms	4.5s
double gyre	45	100×100	5.7s	0.2ms	57.3s
pendulum	25	20×20	0.2s	0.1ms	162.8s

Table 1. Timings. For each data set the table shows integration time τ, seed grid resolution, and the computation times for each phase of our algorithm: backward integration in $\mathbf{w}$, candidate selection, and ridge sampling.

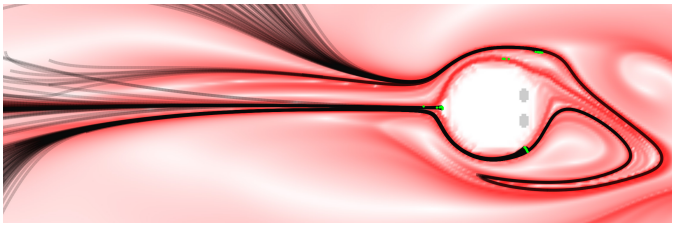

Fig. 12. Flow around a circular cylinder over integration time $\tau = 150$. Ridges are integrated for every point integrated backwards in $\mathbf{w}$.

Figure 12 shows a close-up of the flow around the cylinder for $\tau = 150$, which is the same setup as in Figure 11 (top left). It contains all clustered points after integration in $\mathbf{w}$ towards $t = 0$. In this case, no candidate selection as described in Section 5.2 has been performed. Instead, ridges were integrated from all points and superimposed. The backward-sloping ridge below the cylinder, which was missed before, now clearly becomes visible. The reason for this difference is the suboptimal choice of cluster representative selected in Figure 11 (top left). Nevertheless, backward integration in $\mathbf{w}$ yields the "correct" separation point trajectory, but it requires intricate selection for non-perfectly converged clusters. This issue has a stronger effect when using our method with short integration times making our

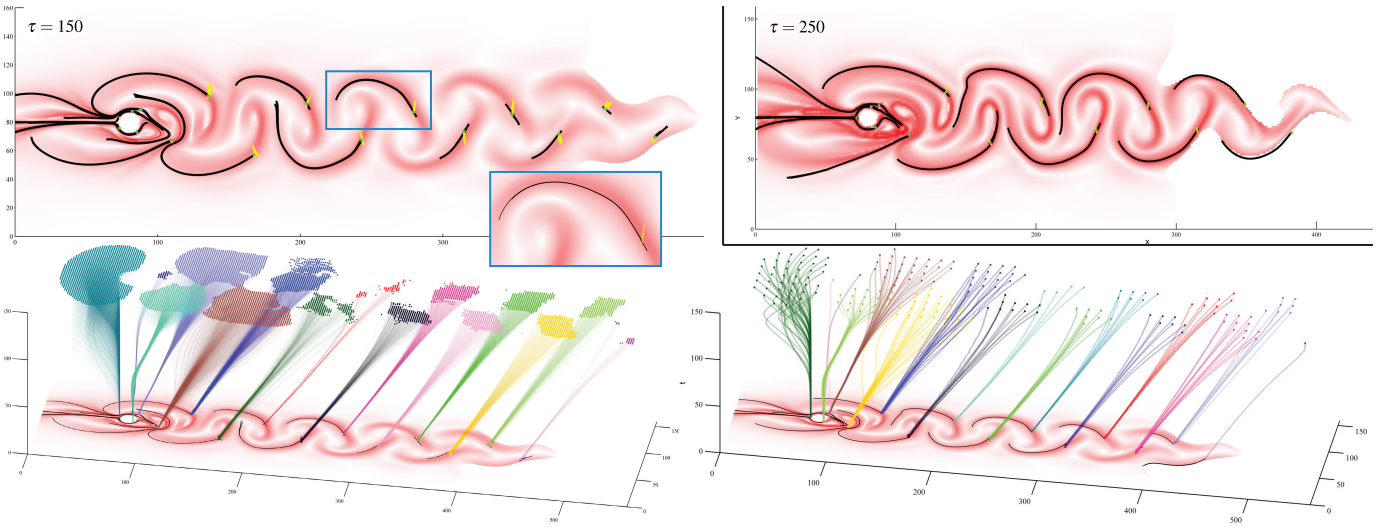

Fig. 11. Results for the PIV-measured flow around a cylinder for $\tau = 150$ (top left) and $\tau = 250$ (top right). FTLE ridges are depicted as black lines. Yellow dots indicate cluster points from integration in the reflection field and their representatives (green). Bottom: Space-time trajectories of reflection lines of a $300\times$ grid (left) and a coarser 50×16 grid (right). In both cases, ridges are obtained as smooth curves.

method not the first choice in such a case. However, the effect quickly alleviates with increasing integration time as indicated by our other experiments (see Figure 9 and 10).

7 Discussion

In the following we discuss our method in comparison to existing methods and summarize advantages and limitations.

Relation to sampling-based FTLE techniques

First of all, our technique is for ridges only! If complete FTLE fields (either as height fields or as color coded image) are desired, our technique is not the right choice. Furthermore, our technique is good at long integration times when the ridge gets very sharp and thin. Since this is the situation where sampling-based techniques have problems due to the required high sampling density, we see our approach not as a replacement but as a complement to sampling-based techniques.

Relation to Sadlo and Weiskopf [23]

Our approach shows some similarities with [23]. In particular, the idea of tracking a ridge from the trajectory of a separation point is similar. However, the way for finding this seed curve is fundamentally different: [23] (and [30]) obtain seed structures from a sampling-based approach, namely by finding and intersecting forward and backward FTLE ridges. Therefore, they carefully elaborate on the sensitivity to the sampling density. Contrarily, our approach needs relatively few integrations to obtain seed curves with high accuracy. This allows us to extract longer and more complicated FTLE ridges than [23] (consider the double gyre data set which is also shown there).

Extension to 3D

The process of creating the reflected field $\mathbf{w}$ can be extended to 3D. In fact, all formulas (2),...,(6) apply to 3D as well except for (5). For this, there exists a 3D extension of feature flow fields [31]. The main technical problem in 3D is the surface sampling and reconstruction from point samples. In particular, there is no simple arc length parametrization as for 1D curves (see 5.3). The efficient and accurate sampling and reconstruction of 2D ridge surfaces that are approximated by uniformly sized or gradually adapting elements (triangles or quads) is much more challenging. We leave this to future research.

Sensitivity of the offset parameter γ

Generally, the larger the offset $|\gamma|$ (see 5.3), the further the extracted ridge goes, but the less accurate it tends to be. The problem with coming up with a reliable offset has been discussed in [23] and [30], and it manifests similarly here: The choice of $|\gamma|$ depends on the data and error of the seed curve. And there is no guarantee that a particular choice ensures tangent continuity of the ridge.

Completeness of the extracted ridges

Our method finds only those FTLE ridges that are caused by moving saddles. Other possible causes of FTLE ridges, such as a strong shear are not detected. We do not see this as a limitation since such ridges are often undesired and removed in a post-process [19] anyway.

8 Conclusions

We have introduced a new approach to extract 2D FTLE ridges by constructing and integrating a new time-dependent vector field that is obtained by reflecting the given velocity field. This way, particles converge toward the ridge under backward integration. Our approach needs only a few samplings and increases accuracy with integration time, not with increasing sampling density. In this sense, we contribute a new method that complements the standard sampling-based FTLE extractors especially for very long integration times.

References

[1] E. Acar, T. Senst, A. Kuhn, I. Keller, H. Theisel, S. Albayrak, and T. Sikora. Human action recognition using lagrangian descriptors. In *IEEE Workshop on Multimedia Signal Processing (MMSP 2012)*, 2012. 2

[2] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4(4):353–373, dec 1994. 1

[3] C. Garth, F. Gerhardt, X. Tricoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1464–1471, 2007. 2

[4] C. Garth, G. Li, X. Tricoche, C. Hansen, and H. Hagen. Visualization of coherent structures in transient 2d flows. In *Proceedings of TopoInVis*, 2007. 2, 5

[5] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D*, 149:248–277, 2001. 1

[6] G. Haller. Lagrangian structures and the rate of strain in a partition of two-dimensional turbulence. *Physics of Fluids*, 13(11), 2001. 2

[7] G. Haller. A variational theory of hyperbolic lagrangian coherent structures. *Physica D: Nonlinear Phenomena*, 240(7):574 – 598, 2011. 1

[8] G. Haller and F. J. Beron-Vera. Geodesic theory of transport barriers in two-dimensional flows. *Physica D: Nonlinear Phenomena*, 241(20):1680 – 1702, 2012. 2

[9] G. Haller and T. Sapsis. Lagrangian coherent structures and the smallest finite-time lyapunov exponent. *Chaos*, 21(2):023115, 2011. 2

[10] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D*, 147(3-4):352–370, Dec. 2000. 1

[11] G. Kindlmann, R. Estepar, S. Smith, and C. F. Westin. Sampling and visualizing creases with scale-space particles. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1415–1424, 2009. 2

[12] A. Kuhn, C. Rössl, T. Weinkauf, and H. Theisel. A benchmark for evaluating ftle computations. In *Proceedings of 5th IEEE Pacific Visualization Symposium (PacificVis 2012)*, pages 121–128, Seoul, South-Korea, March 2012. 1, 2, 5

[13] A. Kuhn, T. Senst, I. Keller, T. Sikora, and H. Theisel. A lagrangian framework for video analytics. In *IEEE Workshop on Multimedia Signal Processing (MMSP 2012)*, 2012. 2

[14] F. Lekien, C. Coulliette, A. J. Mariano, E. H. Ryan, L. K. Shay, G. Haller, and J. Marsden. Pollution release tied to invariant manifolds: A case study for the coast of Florida. *Physica D: Nonlinear Phenomena*, 210(1-2):1–20, 2005. 2

[15] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. In *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, CVPR '96, pages 465–, Washington, DC, USA, 1996. IEEE Computer Society. 1

[16] D. Lipinski and K. Mohseni. A ridge tracking algorithm and error estimate for efficient computation of lagrangian coherent structures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1), 2010. 2

[17] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics*, 23(3), 2004. 2

[18] R. Peikert and F. Sadlo. Height ridge computation and filtering for visualization. In *PacificVis*, pages 119–126. IEEE, 2008. 1, 2

[19] A. Pobitzer, R. Peikert, R. Fuchs, H. Theisel, and H. Hauser. Filtering of ftle for visualizing spatial separation in unsteady 3d flow. In R. Peikert, H. Hauser, H. Carr, and R. Fuchs, editors, *Topological Methods in Data Analysis and Visualization II*, pages 237–253. Springer, 2012. 7

[20] F. Sadlo and R. Peikert. Efficient visualization of lagrangian coherent structures by filtered amr ridge extraction. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2007)*, 13(6):19–26, 2007. 2

[21] F. Sadlo and R. Peikert. Visualizing lagrangian coherent structures and comparison to vector field topology. In *Proceedings of the 2007 Workshop on Topology-Based Method in Visualization (TopoInVis)*, 2007. 2

[22] F. Sadlo, A. Rigazzi, and R. Peikert. Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, editors, *Topological Methods in Data Analysis and Visualization*, Mathematics and Visualization, pages 151–165. Springer, 2009. 2

[23] F. Sadlo and D. Weiskopf. Time-Dependent 2D Vector Field Topology: An Approach Inspired by Lagrangian Coherent Structures. *Computer Graphics Forum*, 29(1):88–100, 2011. 2, 7

[24] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and strain skeletons in eulerian and lagrangian frames. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):980–990, 2007. 1

[25] B. Schindler, R. Peikert, R. Fuchs, and H. Theisel. Ridge concepts for the visualization of lagrangian coherent structures. In R. Peikert, H. Hauser, H. Carr, and R. Fuchs, editors, *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pages 221–235. Springer Berlin Heidelberg, 2012. 2

[26] T. Schultz, H. Theisel, and H.-P. Seidel. Crease surfaces: From theory to extraction and application to diffusion tensor mri. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):109–119, 2010. 2

[27] 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: Nonlinear Phenomena*, 212(3-4):271–304, 2005. 1, 2

[28] S. C. Shadden, F. Lekien, J. D. Paduan, F. P. Chavez, and J. E. Marsden. The correlation between surface drifters and coherent structures based on high-frequency radar data in monterey bay. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3-5):161 – 172, 2009. 2

[29] H. Theisel and H.-P. Seidel. Feature flow fields. In G.-P. Bonneau, S. Hahmann, and C. H. (Ed.), editors, *Proc. Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym'03)*, pages 141–148, 2003. 3

[30] M. Üffinger, F. Sadlo, and T. Ertl. A time-dependent vector field topology based on streak surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):379–392, 2013. 2, 7

[31] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Feature flow fields in out-of-core settings. In *Proc. Topo-In-Vis 2005*, pages 51–64, 2007. 7

[32] M. Weldon, T. Peacock, G. B. Jacobs, M. Helu, and G. Haller. Experimental and numerical investigation of the kinematic theory of unsteady separation. *Journal of Fluid Mechanics*, 611, 2008. 2