

# A Time-Dependent Vector Field Topology Based on Streak Surfaces

Markus Üffinger, Filip Sadlo, *Member, IEEE*, and Thomas Ertl, *Member, IEEE*

**Abstract**—It was shown recently how the 2D vector field topology concept, directly applicable to stationary vector fields only, can be generalized to time-dependent vector fields by replacing the role of stream lines by streak lines [1]. The present paper extends this concept to 3D vector fields. In traditional 3D vector field topology separatrices can be obtained by integrating stream lines from 0-dimensional seeds corresponding to critical points. We show that in our new concept, in contrast, 1-dimensional seeding constructs are required for computing streak-based separatrices. In analogy to the 2D generalization we show that invariant manifolds can be obtained by seeding streak surfaces along distinguished path surfaces emanating from intersection curves between codimension-1 ridges in the forward and reverse finite-time Lyapunov exponent (FTLE) fields. These path surfaces represent a time-dependent generalization of critical points and convey further structure in time-dependent topology of vector fields. Compared to the traditional approach based on FTLE ridges, the resulting streak manifolds ease the analysis of Lagrangian coherent structures (LCS) with respect to visual quality and computational cost, especially when time series of LCS are computed. We exemplify validity and utility of the new approach using both synthetic examples and computational fluid dynamics results.

**Index Terms**—Vector field topology, Lagrangian coherent structures, streak lines, time-dependent vector fields.

## 1 INTRODUCTION

FEATURE extraction is one of the primary means for analyzing large and intricate vector fields. A prominent feature concept is *vector field topology* providing the overall structure of vector fields by means of selected stream lines. It depicts the qualitative transport behavior in steady vector fields by extracting the so-called *separatrices*, sets of stream lines that separate regions of qualitatively different transport behavior. They are obtained by selecting all stream lines that converge to saddle-type *critical points* in forward or reverse time. Critical points can be regarded as degenerate stream lines, i.e., isolated stream lines consisting of a single point—or, in other words, isolated points where the vector field has zero velocity. They are classified by the vector field behavior in their vicinity by means of the velocity gradient. The type *saddle*, also called *hyperbolic* and of central interest in our case, exhibits both positive and negative real parts of the velocity gradient eigenvalues. The vector field topology concept includes further special types of stream lines such as closed stream lines called periodic orbits; these are, however, not present in the context of aperiodic time-dependent flow, the topic of this paper.

Visualization based on vector field topology looks back on a long history and has proven successful in a multitude of domains and applications in 2D as well as in 3D. There is, however, a major drawback with the approach: it is—in a strict sense—limited to stationary vector fields. The reasons are the instantaneous nature of stream lines (they are integral curves in “snapshots” of the vector field) and the fact that both critical points and separatrices are defined as limit cases as integration

time goes to  $\pm\infty$ . Although vector field topology is sometimes used for the investigation of time-dependent vector fields, e.g., if time-dependency is comparably small, it is clear that integrating inside snapshots of the vector field for arbitrarily long times cannot provide insight in the true time-dependent transport phenomena in general. Additionally, typical vector fields from simulation are given on finite time domains and hence the aforementioned limit cases are not available. In the last decade the alternative approach of obtaining geometric representations of time-dependent separatrices, also called Lagrangian coherent structures (LCS), by ridge extraction from the finite-time Lyapunov exponent (FTLE) field has become popular [2]. The FTLE is a scalar field computed from forward or reverse trajectories and represents their divergence over a finite time interval. However, this approach lacks a direct connection to critical points and their relation to separatrices, and faces the difficulty of FTLE ridge extraction [3].

We have recently shown [1] that the aforementioned limitations can be fixed in the 2D vector field topology concept by replacing the role of stream lines by streak lines. In the resulting time-dependent vector field topology, saddle-type critical points represent degenerate *streak* lines, i.e., streak lines consisting of a single point resulting from seeds that move along *hyperbolic trajectories*, and separatrices represent streak lines converging toward those in forward or reverse time, consistent with FTLE ridges. Besides revealing the true transport in time-dependent vector fields, it represents a generalization of the traditional 2D vector field topology as streak lines are identical to stream lines in stationary fields.

The extension of the 2D concept [1] to 3D vector fields involves several conceptual and algorithmic issues which are subject to this paper. Our contributions include:

- We motivate and show the utility of intersection curves between codimension-1 ridges in the forward and

• M. Üffinger, F. Sadlo, and T. Ertl are with the Visualization Research Center, University of Stuttgart (VISUS), Allmandring 19, 70569 Stuttgart, Germany. E-mail: {ueffinger,sadlo,ertl}@visus.uni-stuttgart.de

codimension-1 ridges in the reverse FTLE field as seeding constructs for our time-dependent streak-based separatrices. Intersections between codimension-1 and codimension-2 ridges turn out to be of limited use in general time-dependent flow. This is in contrast to traditional 3D vector field topology where the counterparts are 1D and 2D manifolds intersecting at saddle-type critical points.

- We present an accurate yet efficient extraction algorithm for ridge intersection curves. It is based on [4] but extracts the intersections directly and is hence more efficient with respect to memory consumption and computation time.
- We provide a seeding strategy for generating our streak manifolds from path surfaces started at the intersections.
- The resulting visualization of streak surface growth provides additional insight in the dynamics of hyperbolic regions, e.g., how they cause LCS.
- We introduce “reverse preadvection”, i.e., reverse advection of the intersection curves prior to streak surface generation. This leads to longer streak generation phases and hence increased (more pronounced) streak manifolds.

Our method shares one limitation with all other topology-related works that are based on ridges in the FTLE field: according to Haller [2], Shadden et al. [5], and [6] these ridges represent LCS only if they exhibit negligible cross-flux, i.e., satisfy the advection principle. Unfortunately, no concise and generic definition has been given so far what “negligible” means in that context. We follow the common practice to require sufficient FTLE advection time and impose a threshold on ridge sharpness based on the Hessian eigenvalues and level of the FTLE, which are chosen by the user in an iterative manner as proposed in [4]. Since a direct validation technique for the cross-flux exists so far for 2D fields only [6], we validate both the seeding path surfaces and their resulting streak manifolds by comparison with respective FTLE ridges.

Although periodic orbits [7] are not directly addressed in this paper, it has to be noted that separatrices of saddle-type periodic orbits that coincide with intersection curves of forward and reverse FTLE ridges are implicitly captured by our approach. However, since time-periodic flows are rather rarely encountered, in particular in practical fluid dynamics, and because a detailed analysis would go beyond the scope of this paper, we address this topic as future work.

The remainder of this paper is organized as follows. Related work with focus on LCS and vector field topology is discussed in Section 2. Section 3 introduces and motivates the concept of our streak-based 3D vector field topology. Section 4 describes our adaptive ridge intersection technique and the streak manifold generation. Results are given in Section 5 for three data sets, an analytical flow and two computational fluid dynamics (CFD) data sets. Section 6 concludes the paper and gives an outlook to future work.

## 2 RELATED WORK

The literature on traditional vector field topology is manifold—there is, however, only little work related to time-dependent vector field topologies. We start with work on traditional vector field topology, continue to work related to time-dependent

alternatives, and conclude this overview with a short view on methods related to streak integration. It is noted here that the present paper does not address the generation of integral surfaces, in particular streak generation. This was, due to its intricacy, subject to intense research in recent years.

The concept of vector field topology was introduced in scientific visualization for 2D [8] and 3D vector fields [9], [10] by Helman and Hesselink, and Globus et al. As introduced, it can be formulated in terms of special stream lines, or in other words, special initial value problems. Beside the aforementioned critical points, separatrices, and periodic orbits in 2D, three-dimensional topology further includes nested tori, and strange attractors. Lines of flow separation and attachment are also closely related. In fact these structures can be obtained as separatrices [11], though also ridge-based approaches [12] and parallel vectors [13] solutions have been presented [14]. The unity of all the invariant structures is understood as the topological skeleton of the vector field. The reader is referred to, e.g., the work of Asimov [15] for further details.

Only few variants of time-dependent topologies have been proposed so far. One of the first was the path line oriented topology by Theisel et al. [16]. More recently, Germer et al. [17] proposed an approach measuring the separation of trajectories integrated through the complete available time domain, resembling the traditional (infinite-time) Lyapunov exponent. Alternative concepts for time-dependent critical points have been investigated by Kasten et al. [18] and Fuchs et al. [19]. We build on the approach by Sadlo and Weiskopf [1] because this definition is closely related to Lagrangian coherent structures (LCS) present as ridges in the finite-time Lyapunov exponent (FTLE) field, currently seen as a powerful alternative for a time-dependent topology. The FTLE was made popular by Haller [2] (see, e.g., [4] for prior works). Interestingly, an early work by Haller proposed to extract invariant manifolds, or LCS, in time-dependent 2D flow by means of *hyperbolicity time* [20]. He obtains *hyperbolic trajectories* by intersecting ridges inside the forward and reverse hyperbolicity time fields and also defines a more restrictive property, the *uniform hyperbolicity* of path lines. Sadlo and Weiskopf reinterpreted this approach as a streak line-based time-dependent vector field topology, generalizing the traditional vector field topology concept to time-dependent vector fields by replacing the role of stream lines by streak lines. Hence, saddle-type critical points are represented by degenerate streak lines, i.e., streak lines generated by seeds moving along hyperbolic path lines, and separatrices represented by streak lines converging toward these path lines in forward or reverse time. They replaced hyperbolicity time by the FTLE field due to advantages in practical flow fields. However, they had problems obtaining uniformly hyperbolic trajectories in 2D flow simulations but have demonstrated that requiring simple hyperbolicity, i.e., both positive and negative real eigenvalues of the Jacobian ( $\det(\nabla \mathbf{u}) < 0$ ) of the flow field  $\mathbf{u}$  allowed them to obtain space-time streak manifolds consistent with LCS in practical flow simulations. In his seminal paper [2] Haller extended his 2D uniform hyperbolicity concept [20] to 3D. Unfortunately, we were not able to successfully apply it to practical flow simulations. So

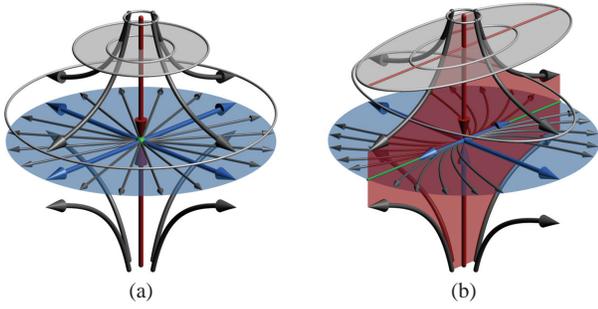


Fig. 1. Saddle-type critical point (green sphere) in vector field  $\mathbf{u}$  with stream lines (gray arrows) and eigenvalue-weighted eigenvectors of  $\nabla \mathbf{u}$  (red and blue arrows). (a) Isotropic (degenerate) case. Contour lines (gray disc) visualize forward FTLE: it has isotropic profile, hence best represented by a 1D ridge (red arrow, consistent with stable manifold). (b) Anisotropic (non-degenerate) case. Forward FTLE has anisotropic profile, hence best represented by a ridge surface (red) oriented along red line in the profile. The green curve represents the intersection of the forward and reverse (blue) FTLE ridge surface.

we follow the approach due to Sadlo and Weiskopf using simple hyperbolicity (we extend it to 3D in Section 4.2.2). We demonstrate in this paper that it is sufficient to require this weak hyperbolicity to obtain LCS by streak manifold generation in practical 3D vector fields.

More recently, Ferstl et al. [21] have presented a streak surface seeding strategy based on cross sections of FTLE fields. They directly use FTLE ridges for seeding streak surfaces and do not aim at a time-dependent vector field topology. We follow an alternative approach, we provide a technique for obtaining hyperbolic path surfaces, the intrinsic seeding structures for streak-based time-dependent 3D topology. These constructs represent a time-dependent generalization of critical points and also convey the space-time structure of this topology: the time-dependent invariant manifolds can be reconstructed by simple streak integration from these structures.

There are several recent related works on the generation of integral surfaces. Garth et al. presented techniques for accurate generation of path and stream surfaces [22], [23]. More recently, Krishnan et al. [24] presented techniques for computing time surfaces and streak surfaces in large data, and Bürger et al. [25] for interactive computation of streak surfaces on the graphics processing unit (GPU). Regarding the streak concept, we use generalized streak lines due to Wiebel et al. [26] for our streak manifolds, i.e., we allow their seeds to move over time. Finally, we would like to refer to the method by Weinkauff et al. [27] for computing traditional streak lines as integral curves inside a derived vector field.

### 3 STREAK-BASED VECTOR FIELD TOPOLOGY

We now describe our extension of the method for 2D vector fields by Sadlo and Weiskopf [1] to 3D vector fields, i.e., we derive a time-dependent 3D vector field topology based on

saddle-type (hyperbolic) sets of path lines, and their separatrices consisting of streak manifolds. Similar to other finite-time concepts, our vector field topology relates to a point in time  $t_0$  and a finite advection time  $T$ , which serves as a scale parameter, i.e., it enables the user to reduce the complexity of the resulting visualizations compared to the traditional Lyapunov exponent. As in the 2D approach, the intersections of ridges in the forward and reverse FTLE fields are used as seeding constructs for the streak manifolds.

The first step (Section 3.1) consists of computing two FTLE fields, one forward and one in reverse advection time, both starting at time  $t_0$ . We indicate those by  $\sigma_{t_0}^T$  and  $\sigma_{t_0}^{-T}$ , respectively. The FTLE is obtained from the flow map  $\phi_{t_0}^T$ , which maps the starting points  $\mathbf{x}$  of trajectories to their respective end points  $\phi_{t_0}^T(\mathbf{x})$  after advection for time  $T$  starting at  $t_0$ :

$$\sigma_{t_0}^T = \frac{1}{|T|} \ln \sqrt{\lambda_{\max} \left( (\nabla \phi_{t_0}^T)^\top \nabla \phi_{t_0}^T \right)} \quad (1)$$

with  $\lambda_{\max}(\cdot)$  being the major eigenvalue. According to Haller [2], ridges [28] in  $\sigma_{t_0}^T$  represent *repelling* LCS and those in  $\sigma_{t_0}^{-T}$  *attracting* ones. The FTLE fields could be computed in a brute force manner, but since high resolution FTLE fields are crucial for obtaining accurate seeding structures and because FTLE computation is expensive, adaptive FTLE evaluation methods such as those by Garth et al. [29], or Hlawatsch et al. [30] should be used for acceleration. Since we require accurate extraction of *ridges* in the FTLE field, we present a technique in Section 4.1 inspired by the acceleration approach by Sadlo and Peikert [4] where a uniform FTLE sampling grid is maintained around the ridges using an AMR approach. Instead of extracting FTLE ridges with subsequent intersection, our technique performs direct refinement toward the ridge intersections. However, to ease description we assume in this section that complete ridges are extracted and then intersected.

In a second step (Section 3.2) the extracted ridge intersections are advected with the flow for time  $T_s$  while producing seeding structures for the streak manifolds. In other words, a path line is generated for each point of the intersections and during integration of the path line particles are continuously released at the front of the path line. These particles represent the space-time streak manifolds. Streak generation is only performed as long as the respective path line is hyperbolic, i.e., particles are only released in hyperbolic regions.

The remaining algorithmic details of our approach are given in Section 4 together with a complexity analysis.

#### 3.1 Seed Generation

In the 2D approach [1], isolated hyperbolic *trajectories* have been identified by *hyperbolic intersections* of ridge lines in the forward and reverse FTLE fields, and used for seeding streak *lines* over time, resulting in 2D *streak manifolds* in space-time. This is consistent with traditional vector field topology in terms of critical points and separatrices as for instants of time, path lines represent points, and the streak manifolds represent lines.

As the concept is extended to 3D, we encounter the question if one should intersect 1D ridges with 2D ridges, or 2D ridges with 2D ridges. Figure 1 illustrates a 3D saddle-type region

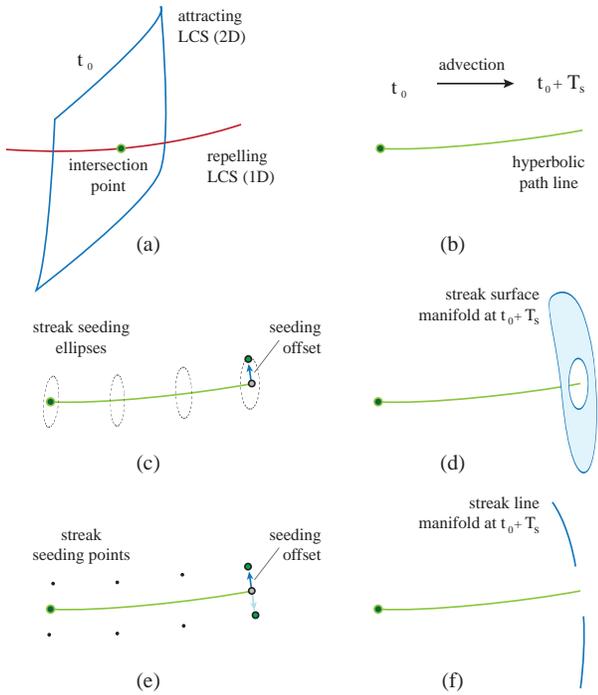


Fig. 2. Streak manifold construction for 1D/2D ridge intersection case. (a) Forward (1D, red) and reverse FTLE (2D, blue) ridges at  $t_0$  are intersected, resulting in ridge intersection point (green). (b) The point is advected for time  $T_s$ , generating a hyperbolic path line. For non-degenerate cases (non-vanishing median eigenvalue of  $\nabla \mathbf{u}$ ) ellipse seeding structures along the path line (c) generate a streak surface manifold (d). In degenerate cases seeding points (e) generate a streak line manifold (f).

where the flow is attracted by the (unstable) 2D manifold and is repelled from the (stable) 1D manifold. In this case two positive and one negative eigenvalues of the Jacobian  $\nabla \mathbf{u}$  encode the hyperbolic behavior (1 in, 2 out) of the flow field  $\mathbf{u}$ —two eigenvectors spanning the surface, and one eigenvector oriented along the 1D manifold, respectively. In the isotropic (degenerate) case the positive eigenvalues are equal (Figure 1(a)), whereas in the anisotropic (practical) case they are not (Figure 1(b)). Hence, extracting ridge *lines* from FTLE is, in a strict sense, an ill-posed problem. Due to the anisotropy in the FTLE profile (gray disc in Figure 1(b)) there is always a 2D ridge that fits better than a respective 1D ridge. Although 1D FTLE ridges have been given no attention so far in the visualization of 3D vector fields, they represent coherent structures in the sense that they tend to exhibit locally highest hyperbolicity [2]. However, like 1D manifolds in traditional 3D vector field topology, they fail at separating regions of different behavior.

We now outline a possible extension of the streak manifold topology [1] to 3D vector fields based on 1D/2D ridge intersection (Section 3.1.1) and discuss its shortcomings. This is followed by 2D/2D ridge intersection (Section 3.1.2) and the motivation for our decision to use 2D/2D intersections throughout our approach (Section 3.1.3).

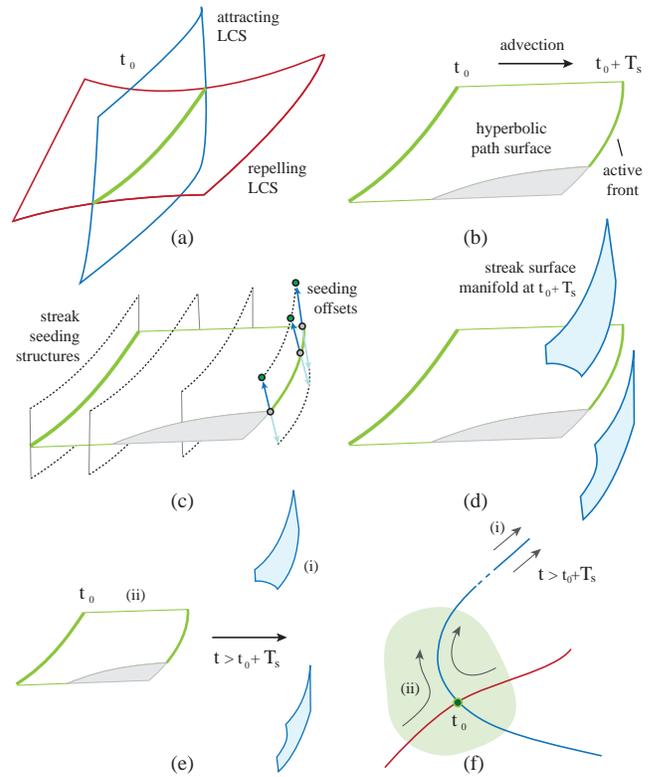


Fig. 3. Streak manifold construction for 2D/2D ridge intersection case. (a) Forward (red) and reverse (blue) FTLE ridges at  $t_0$  are intersected, resulting in a ridge intersection curve (green). (b) The curve is advected for time  $T_s$ , resulting in a hyperbolic (green) path surface. (c) During integration, two seeding curves are generated by offsetting from the path surface front in direction of the major eigenvector (blue) of  $\nabla \mathbf{u}$ . (d) Resulting streak manifolds on either side of the hyperbolic path surface. (e) Streak surfaces can be advected even after generation has stopped, corresponding to remote space-time relation between (i) and (ii) in traditional vector field topology (f).

### 3.1.1 1D/2D Ridge Intersection

With 1D/2D ridge intersection one can extract the 1D ridges either from the forward or from the reverse FTLE field (and the 2D ridges vice versa). Although both results in the same intersection points in simple cases (Figure 4(c) and (d)), it might affect the resulting intersections in complex cases. In the following discussion we restrict ourselves to the (1 in, 2 out) case (Figure 1, the opposite case is obtained by reversal of the vector field). In a first step FTLE ridges are extracted from the forward and reverse FTLE field at time  $t_0$  to obtain repelling and attracting LCS (Figure 2(a)). The 1D ridges are extracted with the parallel vectors approach [13] while the 2D ridges are formulated according to [4]. The intersection points are then advected for time  $T_s$  and the initial portion of the resulting path lines residing in hyperbolic regions are the *hyperbolic path lines* (Figure 2(b)) which would serve for seeding structures for streak manifold generation. As in the case of traditional vector field topology and its time-dependent variant [1], the seeding points for the streak lines have to be slightly offset from the

path line to allow the streaks to grow. A common choice is to apply an eigen analysis of  $\nabla \mathbf{u}$  which reflects the local flow behavior. In the isotropic case (equal positive eigenvalues), one could simply generate a seed circle spanned by the major eigenvectors to obtain a streak surface that coincides with the attracting LCS (Figure 2(c) and (d)). In anisotropic cases where the medium eigenvalue is smaller but nonzero, a seeding ellipse corresponding to the major and medium eigenvalues (-vectors) would produce a streak surface consistent with the attracting LCS. However, if the medium eigenvalue is comparably small (strong anisotropy), the seeding circle would degenerate to a line and hence two seeds, offset along the major eigenvector, would produce the same line-type subset of the desired attracting LCS (Figure 2(e) and (f)).

### 3.1.2 2D/2D Ridge Intersection

For the 2D/2D case, ridge *surfaces* are extracted from both forward and reverse FTLE, with subsequent ridge intersection, see Figure 3(a). The resulting intersection curves are advected for time  $T_s$  resulting in path surfaces and their parts within hyperbolic regions constitute the *hyperbolic path surfaces* (green in Figure 3(b)). The hyperbolic path surfaces take the role of critical points: they are used for seeding the streak manifolds at their front, see Figure 3(c) and (d). Again, eigensystem-based offsetting is applied to generate the seeding curves for streak surface generation, described in Section 4.2.2.

### 3.1.3 Discussion

Figure 4 demonstrates the problems of 1D/2D ridge intersection in the simple synthetic Gyre-Saddle data set (Section 5.1) with anisotropy 2 : 1, which is below anisotropy in typical CFD data sets. Several stray 1D ridges are obtained (Figure 4(c) and (d)). In contrast, 2D/2D ridge intersection is typically more robust (Figure 4(b)), especially for FTLE fields from CFD data. Figure 4(e) and (f) highlight the main advantage of the 2D/2D ridge intersection approach in anisotropic configurations: With the seeding *point* resulting from 1D/2D ridge intersection, only a small elongated streak surface is obtained which covers only a small fraction of the corresponding LCS at time  $t_0 + T_s$  (Figure 4(e)). Please note that using infinite advection time, as used in traditional vector field topology to allow the manifold to grow to the desired size also in “perpendicular direction”, is usually not possible in time-dependent vector fields due to limited time domain and temporal variation. The seeding *line* resulting from the 2D/2D approach instead, generates streak surfaces that capture the LCS much better within the available time scope (Figure 4(f)). All this motivates the 2D/2D ridge intersection approach.

## 3.2 Streak Manifold Generation

Our method employs seeding curves from 2D/2D ridge surface intersection for streak manifold generation as motivated in Section 3.1. Two offset seeding structures generate the two parts of the streak manifold at the front of each hyperbolic path surface during integration time  $T_s$  (Figure 3(c) and (d)).

The growth of the streak surfaces depends on the hyperbolicity (repulsion) along the hyperbolic path surface and the

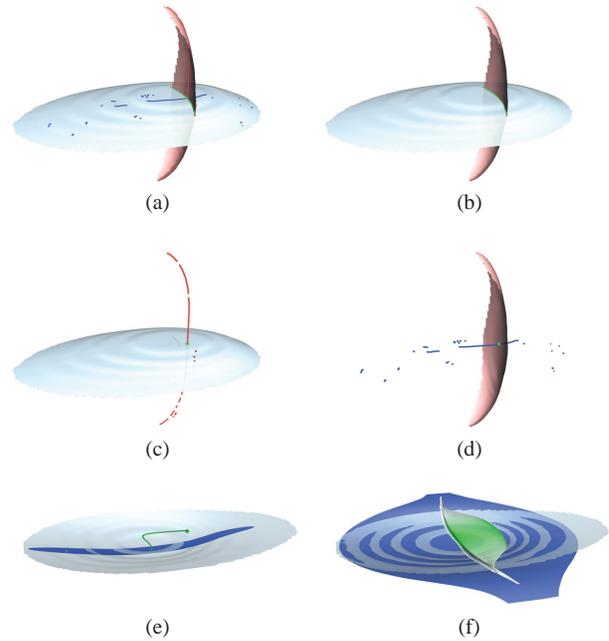


Fig. 4. Streak surface generation with 1D/2D vs. 2D/2D ridge intersection in Gyre-Saddle example. (a) 1D and 2D forward (red) and reverse FTLE ridges (blue) allow three intersection cases: 2D/2D ridge intersection with resulting curve (green) (b), intersection point from 2D reverse and 1D forward FTLE ridges (c), and reverse case (d). (e) Streak manifold strip (dark blue) from 1D/2D intersection point together with matching FTLE ridge (light blue) and path line (green). (f) Superior streak surface (dark blue) from 2D/2D ridge intersection with path surface (green).

advection time  $T_s$  during which it acts on the streak surface. We introduce *reverse preadvection*, to allow for increased  $T_s$  and thus larger streak manifolds for more significant visualizations. The basic idea is to advect the ridge intersection curves in reverse-time direction for time  $T_p$  prior to streak generation (Figure 16(c)). To simplify the streak generation stage and to allow the interpretation of hyperbolicity by the shape of the resulting streaks (Figure 13), we limit  $T_p$  such that the curves stay completely inside hyperbolic regions. Starting at the new space-time location of the seeding curve, streak generation for additional time  $T_p$ , provides larger streak manifolds (Figure 16(b), (d)).

### 3.2.1 Discussion

As in the underlying 2D approach [1] the time  $T_s$  during which a streak manifold can be robustly generated (i.e., results in geometry consistent with LCS) is limited by several factors. Note that preadvection time  $T_p$  does, except for numerics, not account in our reasoning because the intersection curve error that grows during reverse advection for time  $T_p$  (preadvection) is subject to annihilation (due to error reduction) during subsequent forward advection for time  $T_p$  as the streaks are generated. As in [1], we primarily require a conservative choice  $T_s \leq T$  with respect to FTLE advection time  $T$ . Secondly,  $T_s$  is limited by the accuracy at which the intersection curve is

extracted. This accuracy depends on the advection property of the ridge surfaces, i.e., on their sharpness [5], [6], and on the FTLE resolution (Section 4.1), both not yet linked to  $T$  by appropriate models. Hence, we reject insufficiently sharp ridges and limit  $T_s$  by comparing the streak manifolds with corresponding FTLE ridges from time to time (Figure 16(g) and (h)). The same applies to the choice of the seeding offset distance (Section 4.2.2) which we choose manually. Note, the offset has to be chosen large enough to allow the streak surface to escape the hyperbolic region and grow. Note also, that the aimed LCS are attracting in the respective time direction. Hence, the streak manifolds are advected toward the LCS and errors are reduced. Finally, streak generation is, as already mentioned, stopped when the respective part of a hyperbolic path surface front enters a non-hyperbolic region.

All in all, streak generation may take place inside  $[t_0 - T_p, t_0 + T_s]$ , provided that the respective part of the hyperbolic path surface front has not entered a non-hyperbolic region. Although streak surface *generation* is constrained to that time period, *advection* of a streak surface can be carried on even if its generation has stopped (Figure 3(e)). The reason is that these streak surfaces still exhibit the property of separatrices: if a particle is seeded on either side of the surface and both the particles and the surface are advected back to the time interval  $[t_0 - T_p, t_0 + T_s]$ , the particles will separate at the hyperbolic region along the streak surface. This is in accordance with the fact that in traditional vector field topology separatrices may be locally indistinguishable from surrounding stream lines ((i) in Figure 3(f))—the relation to the hyperbolic region (ii) is remote in space-time, as in our case. To prevent clutter, one can, nevertheless, decide to limit the advection time after streak generation has stopped to an appropriate value. In the shown results we stopped advection at time  $t_0 + T_s$ .

Please also note that our approach performs streak surface integration within this time interval, i.e., it does not produce a single result but the streak integration itself visualizes the dynamics of LCS within the time interval, in contrast to FTLE ridges, which visualize only an instant of time. Time series visualization of the advected and growing streak surfaces provide additional information within the space-time region of interest. One example is the amount of growth of the streak surfaces which directly reflects the action of hyperbolicity.

As in the case of traditional vector field topology, separatrices, and hence also our streak manifolds, are computed in both forward and reverse time. The application of our complete approach in reverse time direction within  $[t_0 - T_s, t_0 + T_p]$  is straightforward, and therefore we constrain our description to forward-time extraction. It has to be noted, however, that time reversal switches the roles of the eigenvectors of the Jacobian  $\nabla \mathbf{u}$  and therefore the computation of the seeding offsets, described in Section 4.2.2, has to be altered accordingly.

## 4 ALGORITHM

The overall algorithm consists of two stages (Figure 5). In a first step (Section 4.1), the intersection curves between the ridge surfaces in the forward and the reverse FTLE field are extracted (Figure 3(a)). Since even comparably small perturbations of the position and shape of the intersection curves

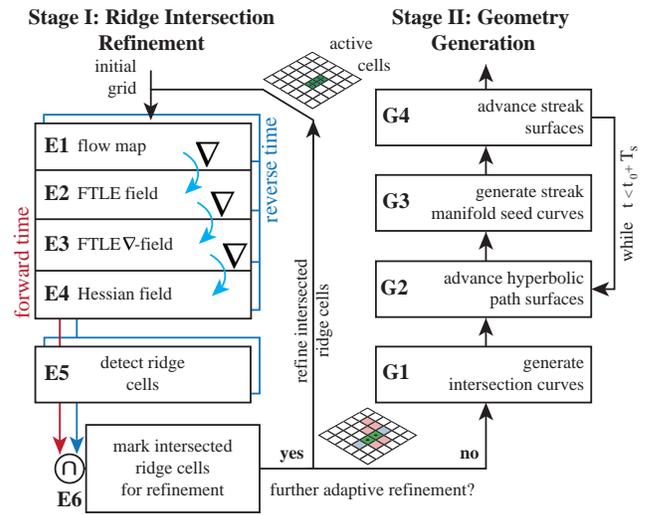


Fig. 5. Data flow pipeline of the overall method. Stage I extracts and refines cells containing potential ridge surface intersections of the forward and reverse FTLE at time  $t_0$ . Stage II derives intersection curve geometry, and advects it for time  $T_s$ , generating the streak manifolds.

can deteriorate the utility of the hyperbolic path surfaces (Figure 15), high resolution FTLE sampling is a prerequisite for achieving the required precision in ridge intersection extraction. FTLE evaluation, however, has a high computational cost because a trajectory needs to be integrated for each sample of the flow map. To reduce the work, we pursue an iterative refinement approach using sparse sampling grid data structures to detect and refine the regions of ridge intersections on a virtual uniform grid. Our method is inspired by the AMR approach by Sadlo and Peikert [4] but follows a sparse multi-grid approach. The method and its differences to [4] are described in Section 4.1.

In a second step the streak manifolds are generated using the previously extracted ridge intersection curves. For this a path surface is seeded from each intersection curve (Figure 3(b)) and two streak surfaces are generated at its front during integration (Figure 3(c) and (d)). The method and its derivation from the 2D approach [1] are described in Section 4.2.

We conclude this section with an investigation of the computational complexity of our method (Section 4.3).

### 4.1 Stage I: Ridge Intersection Refinement

As in the approach due to Sadlo and Peikert [4], the concept central to our ridge intersection refinement are *ridge cells*. A ridge cell is a cell of our FTLE sampling grid that has at least one of its edges intersected by an FTLE ridge surface. According to the criterion by Eberly [28] such an edge exhibits a zero crossing of the derived quantity  $e = \nabla \sigma \cdot \mathbf{e}_0$ , with  $\sigma$  being the FTLE and  $\mathbf{e}_0$  the minor eigenvector (with the additional condition of negative minor eigenvalue) of the FTLE Hessian. To suppress noise and constrain the visualization to ridges that represent LCS (ridges that are sufficiently sharp), we additionally require a minimum FTLE value  $\tau_\sigma$  and sufficient modulus of the minor eigenvalue of the Hessian  $\tau_H$ , according

to [4]. Having ridge cells in the forward and in the reverse FTLE field, the conjunction of these sets represents the set of *intersected ridge cells*. These cells, which possibly contain the ridge intersection curves, are refined by subdivision and then represent the set of *active cells* for the next refinement pass (Figure 5). Since the gradient estimation and Hessian estimation operators require additional support, we maintain a neighborhood of vertices around each active cell for our computations (Figure 6).

The main difference to the approach [4] is that instead of extracting ridge surfaces in isolated FTLE fields, only the intersection between ridges in the forward and reverse FTLE field is refined for. This results in a substantial speedup and memory conservation (Section 4.3). To speed up the tracing of the still large number of FTLE path lines the advection is performed on the GPU using a fourth order Runge-Kutta integrator implemented in CUDA. Another difference is that we avoid the difficulties coming with AMR data structures and instead follow a multigrid approach, i.e., we maintain several Cartesian grids that differ by a power of 2 in cell size. These grids are stored in a sparse manner and only the finest level is kept during refinement, i.e., only cells are represented that are needed during refinement.

The process starts with an initial FTLE evaluation on a comparably coarse uniform grid inside the region of interest. In our experiments we have chosen a resolution that oversamples the grid of the underlying vector field by a factor of two, although other resolutions are possible, of course. If a too low initial sampling resolution is chosen, ridge intersections can be missed. This issue is common to most adaptive schemes, including those for FTLE computation, e.g., [4], [29]. Although our GPU prototype is restricted to vector field input on uniform grids, it can be easily extended to vector field input on unstructured grids. Only the integration of path lines and the estimation of the vector field Jacobian have to be adapted accordingly, e.g., by using the approach due to Garth and Joy [31]. In the following we describe the execution of the pipeline with ridge intersection taking place at time  $t_0$  and subsequent forward-time streak manifold generation from time  $t_0$  to  $t_0 + T_s$ . The algorithmic steps for extracting ridge intersections are described next. They are executed once for forward, and, once for reverse FTLE (Figure 5, E1–E5).

For each refinement level, first the flow map is computed at the new grid vertices (Figure 5, E1). Then the FTLE (Eq. 1) is computed from the flow map (E2). To perform ridge detection according to Eberly [28] on the edge level of the grid, the gradient of the FTLE is computed at the active grid vertices (E3) and the Jacobian operator is applied once more to obtain the Hessian of the FTLE field (E4). Note, the Hessian could also be computed directly from the FTLE field. The indirect way however allows for a straightforward implementation of the refinement algorithm on the sparse grid. Since eigenvectors lack orientation, and to make sure that edge intersections are consistent for all cells adjacent to an edge, we consistently orient the two instances  $\mathbf{e}_0'$  and  $\mathbf{e}_0''$  of the minor Hessian eigenvector at the two end points of the edge (assuring  $\mathbf{e}_0' \cdot \mathbf{e}_0'' \geq 0$ ), detect the ridge cells (E5), and subsequently intersect the cells (E6). Figure 6 illustrates the

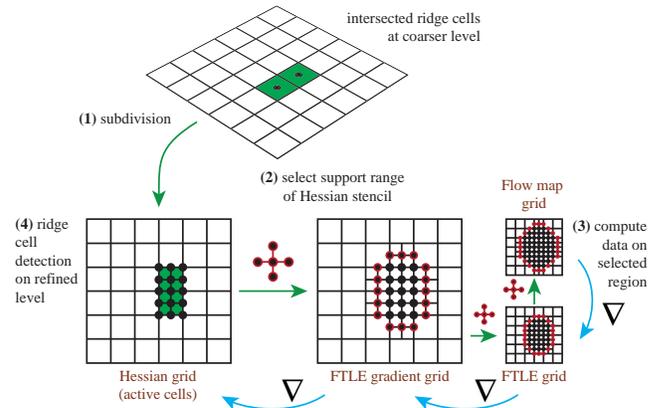


Fig. 6. Adaptive refinement of ridge cells. For clarity a 2D grid and a simple gradient stencil is shown. The green path shows the selection of the neighborhood with respect to the stencil of the active cells (green) in the Hessian grid. Data required for ridge cell detection on the refined level is computed for forward and reverse FTLE (blue path).

refinement of the intersected ridge cells and the selection of the required neighborhood in reverse computation order (green path) to account for the computation operator stencils on the refined level (blue path, E1–E5).

In our experiments we address the FTLE underestimation refinement issue [4] by lowered FTLE thresholds. Other strategies, such as the look ahead approach [4], could be used as well. Similar to [4] we mark cells adjacent to ridge cells to enable detection of new ridge cells due to improving sampling resolution. This allows the set of intersected ridge cells to grow at the ends of the intersection curves during refinement. The ridge intersection curve geometry is finally derived from the resulting set of intersected ridge cells, as described next.

## 4.2 Stage II: Geometry Generation

The input to this stage is the set of intersected ridge cells from Section 4.1 which possibly contain ridge intersection curves. The algorithm first proceeds with intersection curve extraction from these data, described in Section 4.2.1. Details on the subsequent process of streak manifold generation from these curves are given in Section 4.2.2.

### 4.2.1 Intersection Curve Extraction

The intersection curves between the ridge surfaces in the forward and reverse FTLE field are determined from the set of intersected ridge cells in a cell-wise manner (G1 in Figure 5). For each of these cells, the ridge surface case is evaluated using Marching Cubes on the Eberly criterion  $e$  (Section 4.1), and the resulting triangles (at most 5 plus 5 from forward and reverse FTLE, respectively) are intersected to obtain the segments that compose the polyline representation of the intersection curves. Since the resulting curves shall serve as seeding curves for hyperbolic path surfaces, we first reject segments that are located outside hyperbolic regions (Section 4.2.2). Having the segments connected, we apply a filtering of the intersection curves by length, i.e., we reject

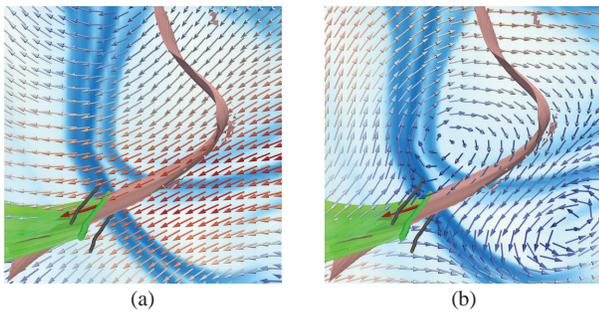


Fig. 7. Eulerian view (a) compared with Lagrangian view (b) of the vector field  $\mathbf{u}$  near a ridge intersection curve (green) (Buoyant Flow data set at  $t_0 = 50.02$  s). The bold red arrow and the hyperbolic path surface (green) depict the motion of the intersection. In the frame of reference of the intersection (b) it is revealed that the two streak surface seed curves (black) should be offset in direction of the major eigenvector of  $\nabla\mathbf{u}$ .

curves below a prescribed length. Filtering features by size is a common procedure in feature extraction to get rid of spurious or insignificant solutions. This allows us to produce LCS visualizations without unnecessary clutter.

#### 4.2.2 Streak Manifold Generation

The streak manifolds are generated by stages G2 to G4 of our pipeline (Figure 5). The ridge intersection curves extracted at time  $t_0$  (Section 4.2.1) are advected with the flow for streak advection time  $T_s$  in an iterative manner, each step producing a new instance of the advancing front of the hyperbolic path surface. In the 2D streak topology approach [1] the streak generating path line is terminated when it enters a non-hyperbolic region, i.e., when  $\det(\nabla\mathbf{u}) \geq 0$ . This includes cases with complex eigenvalues of  $\nabla\mathbf{u}$ . We apply a corresponding filtering to our path surface front: those parts exhibiting complex eigenvalues of  $\nabla\mathbf{u}$  or real eigenvalues of equal sign (non-hyperbolic case) are stopped, i.e., no further path surface is generated therefrom (see Figure 13). Two streak surfaces are generated from each remaining path surface front part, as described in Figure 3. The fronts of the hyperbolic path surfaces cannot be directly used as seeding structures for the generalized streak surfaces. This would simply reproduce the advected seed curve, i.e., the streak surface would degenerate to its seeding curve moving along the hyperbolic path surface. Conceptually, the vertices of the front need to be offset to both sides of the repelling LCS to allow the streak surfaces to escape the hyperbolic region, as illustrated in Figure 7. Hence, we use the major eigenvector of the Jacobian  $\nabla\mathbf{u}$  as offsetting direction, as shown in Figures 3 and 7.

Note that the offset vectors have to be oriented consistently in space and time for both seeding curves to prevent the two parts of the streak manifold flipping from one side to the other side of the repelling LCS during their generation, which would produce severe artifacts. Figure 8(a) illustrates how consistency is achieved. Eigenvector orientation at the first vertex at time  $t_0$  is arbitrarily chosen (bold blue arrow, 0) and made consistent in space only by propagating the orientation

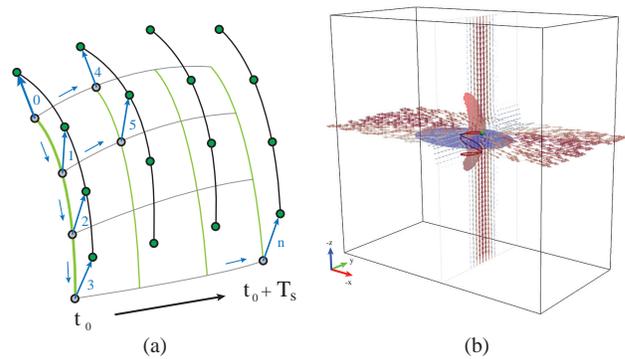


Fig. 8. (a) Offsetting seed curves (black) at the front of the hyperbolic path surface (green). (b) Gyre-Saddle example at  $t = 3.5$  s. The saddle-type critical point (green) of the field translates along a Lissajous curve (red curve).

along the front of the path surface (green curve), e.g.,  $0 \rightarrow 1$ . At later advection instances the offset vector orientations are chosen consistently with their direct predecessors in time, e.g.,  $0 \rightarrow 4$ . Note that the offset vectors could also be tangent to the path surface. Still our approach produces consistent streak manifolds in such configurations. In our current approach we keep the length of the offset vector constant during the whole generation process. With this choice the growth of the streak manifolds directly visualizes their hyperbolic strength, enabling a quantitative comparison of the different hyperbolic regions, as shown in Figure 11. As an alternative, to close the offset gap at the end of the streak generation process, one could let the length of the offset vectors drop to zero as approaching  $t_0 + T_s$ . This, however, would not reveal the seeding offset in the visualization and therefore we did not apply this strategy. Regarding the quality of the streak surfaces we achieved high quality results with our prototype. Nevertheless, the streak surface integration part could be replaced with an adaptive streak surface integration algorithm, like, e.g., by Krishnan et al. [24], to alleviate potential issues in problematic field regions, such as regions with strong divergence  $\nabla \cdot \mathbf{u}$ .

#### 4.3 Complexity Analysis

In the following we analyze the computational complexity of our algorithm and compare it to the traditional approach of obtaining LCS (time series) as ridges from FTLE fields.

For the complexity of the ridge intersection curve extraction we assume a regular grid with resolution  $N^3$ . The straightforward approach for computing a single FTLE instance on a regular grid has complexity  $O(N^4)$ —the cost of the flow map path lines  $O(N)$  times the resolution  $O(N^3)$ .

Our adaptive approach starts with a coarse base resolution of  $M^3$ , with  $M \ll N$ , and then, during refinement, focuses its computational effort on 1D subregions of the region of interest, which potentially contain ridge intersection curves. With  $k$  refinement iterations, and  $M$  chosen such that  $2^k M = N$ , our method has cost  $O(M^3 N + \sum_{i=1}^k 2^i M N)$ . This equals  $O(M^3 N + 2(N - M)N)$ , and with  $k \rightarrow \infty$  the ridge intersection stage has complexity  $O(N^2)$ . This is backed up by our measurements provided in Section 5.1.2. There, of course,  $k$  is limited and

the operations on the sparse grid during refinement have a larger constant factor, as revealed in Table 1. In contrast, a similar complexity analysis yields  $O(N^3)$  for the adaptive ridge surface extraction of Sadlo and Peikert [4].

For the second stage of our approach, the streak surface generation from ridge intersection curves, the size of the intersection curves is in the order of  $O(N)$ . Since streak lines exhibit complexity  $O(N^2)$ , our streak surfaces have complexity  $O(N^3)$ , leading to an overall complexity of  $O(N^3)$  for our approach. Nevertheless, our method delivers the dynamics of hyperbolic LCS within the whole time interval  $[t_0 - T_p, t_0 + T_s]$  at no extra cost, i.e., the streak generation process already represents a visualization of LCS dynamics within  $[t_0 - T_p, t_0 + T_s]$ . Extracting corresponding time series with the traditional FTLE ridge-surface approach has, in contrast, complexity  $O(N^5)$ , and  $O(N^4)$  for the adaptive ridge extraction [4], respectively.

## 5 RESULTS

We show results of our method at the example of three time-dependent data sets. First we investigate the synthetic Gyre-Saddle data set, containing a single prominent hyperbolic region. It enables us to exemplify the different aspects of our method, which would be harder to achieve with CFD data. Please see also the accompanying video as an introduction. We then apply our method to a von Kármán vortex street CFD simulation. Lastly, a buoyant flow with a complex LCS structure is investigated. Reverse preadvection has been applied only for the vortex street example. In cases where it was sufficient to employ a single filtering threshold (minimum FTLE  $\tau_\sigma$  or maximum minor Hessian eigenvalue  $\tau_H$  only) we omit the respective value of the other filter ( $\tau_H = 0$  or  $\tau_\sigma = -\infty$ ).

Timings are given for a workstation with a quad-core Intel CPU at 2.67 GHz, and an NVIDIA Fermi 480 GTX GPU.

### 5.1 Time-Dependent Gyre-Saddle

The synthetic Gyre-Saddle field (Section 5.1.1) serves as a means to illustrate the concepts and results of our method at a simple saddle-type region. A detailed analysis of the timings and the corresponding problem sizes on the different refinement levels is given in Section 5.1.2. Moreover, the simple FTLE ridge structure of this data set lends itself well to demonstrate the superior accuracy of our streak manifolds compared to the traditional visualization by FTLE ridges (Section 5.1.3).

#### 5.1.1 Data Set Description and Experiment Setup

The straightforward choice for a simple hyperbolic region would be a linear saddle field. However, for a Lagrangian (Galilean-invariant) concept such a field is futile: it exhibits uniform Jacobian and hence, e.g., a uniform FTLE field without ridges. Therefore, we have constructed a 3D variant of the 2D example presented in [1], exhibiting a cosine velocity profile. The symmetric (zero skew) 2D saddle therefrom is revolved around the central  $z$ -axis, and afterward it is made anisotropic by multiplying its component in  $x$ -direction by a factor of 2 and choosing an extent of

$[-2.5, 2.5] \times [-1.25, 1.25] \times [-2.5, 2.5] \text{ m}^3$ . From this we construct a time-dependent field by translating the region in terms of  $t$  along the Lissajous curve

$$\mathbf{x}(t) = \begin{pmatrix} \cos(5t)/4 \\ \sin(5t)/8 \\ \cos(\frac{5}{3}t)/4 \end{pmatrix}. \quad (2)$$

The time-dependent field is discretized on an equidistant grid at resolution  $51 \times 26 \times 51$  within the time interval  $[0, 10] \text{ s}$ , at regular sampling of  $\Delta t = 0.0125 \text{ s}$ . Figure 8(b) sketches the field at  $t = 3.5 \text{ s}$  together with the Lissajous curve, and FTLE ridges (red - forward, blue - reverse). In terms of traditional vector field topology, there is a stable 1D manifold in  $z$ -direction and an unstable 2D manifold in  $x$ - and  $y$ -direction.

In the following we consider the case with  $t_0 = 3.5 \text{ s}$ , FTLE advection time  $T = 2 \text{ s}$ , and streak generation time  $T_s = 2 \text{ s}$ . We start with an initial FTLE resolution of  $81 \times 41 \times 81$  at refinement level 0 in the region  $[-2, 2] \times [-1, 1] \times [-2, 2] \text{ m}^3$ , and use 4 iterations to refine the ridge intersection curve during Stage I of our algorithm. During refinement we set the FTLE filter to  $\tau_\sigma = 1.0$ , and restrict it to 1.3 on the final level as described at the end of Section 4.1. A streak seeding offset of 0.01 m is chosen. Figure 10 shows some of the results.

#### 5.1.2 Performance and Cost Analysis

Table 1 shows timings and statistics for the four refinement levels. The extraction of the ridge intersection cells on refinement level 4 required 41.4 s in total, with 25 s being allotted to the extraction at refinement level 0, for which, in contrast, the computation steps E1 to E6 need to be performed for all cells of the initial grid (256000 active cells using 31.4 MB for all quantities). It is apparent that our technique strongly benefits from the reduction of the ridge intersection problem from  $O(N^4)$  to  $O(N^2)$  during refinement. The level 0 grid yields 48 potential ridge cells, resulting in 384 refined cells which are processed on the following level. It can also be seen that our algorithm accounts for growing ridges due to increased flow map resolution, e.g., level 2 starts with 856 input cells but detects 910 ridge surface cells in forward FTLE. The more restrictive FTLE filter threshold on the last level is also noticeable. The refinement levels exhibit relatively long execution times compared to the base level with respect to the number of active cells. This is caused by the overhead of the underlying sparse grid data structure. The very costly computations of the flow map, the path surfaces, and the streak

TABLE 1

Gyre-Saddle data set: Timings and problem sizes on different ridge intersection refinement levels starting with a flow map resolution of  $81 \times 41 \times 81$  on level 0 and ending with a virtual resolution of  $1296 \times 656 \times 1296$  on level 4.

refinement level	0	1	2	3	4	total
active cells	256000	384	856	1832	5772	264844
2D ridge cells, bck	1481	342	751	1756	4764	-
2D ridge cells, fwd	908	385	910	2168	2365	-
intersec. ridge cells	48	107	229	659	374	-
t in s	25	1.1	2.0	3.8	9.6	41.4

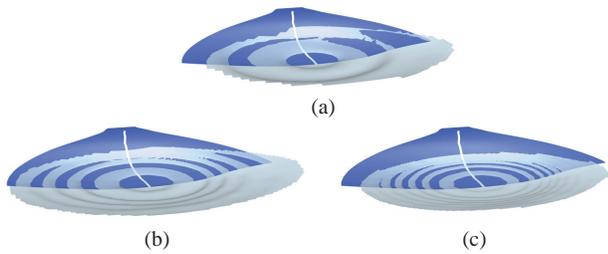


Fig. 9. Superior accuracy of streak manifold (dark blue, cut in the middle) in comparison to reverse time FTLE ridge (light blue). The ridges show aliasing artifacts. With increasing FTLE resolution,  $81 \times 41 \times 81$  (a),  $161 \times 81 \times 161$  (b),  $321 \times 161 \times 321$  (c), the ridges converge toward the streak manifold (Gyre-Saddle data set,  $t_0 = 3.5$  s,  $T_s = 2$  s).

manifolds are executed in parallel on the GPU. Currently, all other stages of our algorithm are implemented on the CPU. In particular, the ridge extraction stage could strongly benefit from a parallel implementation, allowing for further speedup.

The geometry generation stages G1 to G4 are less time consuming, also due to GPU-accelerated streak generation. The advection of the single ridge intersection curve found in the data set, consisting of 651 vertices, for time  $T_s$  with a time step of  $\Delta t = 0.05$  s required 0.4 s, and with streak manifold generation enabled, 11.1 s, resulting in a total time of 52.5 s.

The final refinement level corresponds to an FTLE resolution of  $1296 \times 656 \times 1296$  with approximately  $10^9$  active input cells. In contrast to a total of only 264844 active cells for the adaptive technique the extraction on such a grid would take several orders of magnitude longer. Moreover, the performance advantage grows further when it comes to the visualization of the time-dependent behavior of LCS, i.e., the computation of traditional time series of FTLE ridges compared to our streak generation (Section 4.3).

### 5.1.3 Streak Manifolds vs. FTLE Ridges

Figure 10 shows the different aspects of our technique. The upper row illustrates the explicitly extracted FTLE ridges on FTLE refinement level 0 at time  $t_0 = 3.5$  s, in the middle of the sequence at  $t_0 = 4.49$  s, and at the end at  $t_0 = 5.5$  s. In the chosen time range the hyperbolic region first moves to the left, and then returns to the right. Note, the camera parameters are kept fixed throughout this figure. The second row depicts the evolution of the path surface (hyperbolicity mapped to saturation) seeded at our refined ridge intersection curve, in context of the FTLE ridges at  $t_0$ . The third row shows the streak manifold generation process during path surface integration. The last row compares the streak manifold with the corresponding attracting FTLE ridge (at  $t_0 = 4.49$  s and  $t_0 = 5.5$  s, respectively). The LCS advection principle [5] is reasonably well fulfilled in the first half of the time interval: the front of the hyperbolic path surface matches the FTLE ridge intersection. Although it starts to deviate in the second half of the time interval, the resulting streak manifold still fits the actual LCS (as detailed in section 5.2 of [1]), resulting in a highly accurate visualization compared to traditional visualization by FTLE ridges, which are inherently limited by

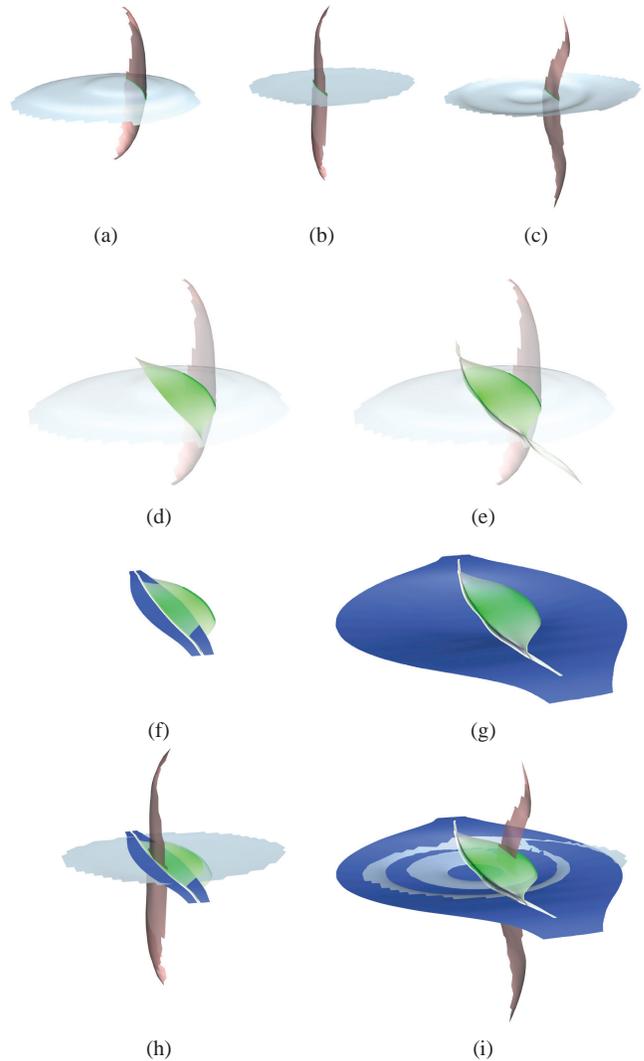


Fig. 10. Streak manifold generation at the example of the Gyre-Saddle data set. First row: Ridge intersection curves (green) of attracting (blue) and repelling FTLE ridges (red) extracted at  $t_0 = 3.5$  s (a), 4.49 s (b), 5.5 s (c), illustrating the dynamics of the hyperbolic region. Second row: Refined ridge intersection curves are advected for  $T_s = 0.99$  s (d), and  $T_s = 2.0$  s (e), producing hyperbolic (green) path surfaces ( $t_0 = 3.5$  s). Simultaneously, streak surfaces (dark blue) are generated at the front of the growing path surface, (f) and (g). Streak manifolds provide a better LCS representation than the attracting FTLE ridge (light blue), (h) and (i), cf. Figure 9.

the FTLE sampling resolution, as demonstrated in Figure 9. With increasing FTLE resolution the ridges converge toward the streak manifold. Please also see the supplementary video.

## 5.2 Von Kármán Vortex Street

The time-dependent CFD flow field of the von Kármán vortex street phenomenon was computed on a domain with extent  $[0, 10] \times [0, 60] \times [0, 10]$  m<sup>3</sup> and a time interval of  $[0, 1.1]$  s. The domain was discretized with an equidistant grid of resolution  $41 \times 241 \times 41$  and a temporal resolution of 0.001 s. A cuboid

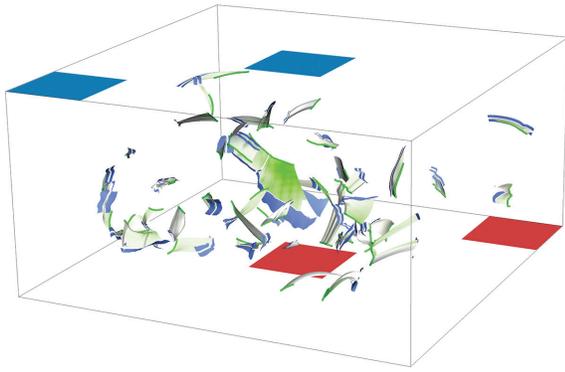


Fig. 11. Streak manifold topology of the Buoyant Flow data set with two heated plates (red) and two cooled plates (blue). The hyperbolicity color mapping on the generating path surfaces highlights the relation between hyperbolicity strength and streak manifold growth ( $t_0 = 50.02$  s,  $T_s = 10$  s, seeding offset 0.025 m, same view as in Figure 12).

obstacle was placed near the inflow region (Figure 16), with inflow velocity in direction of the largest extent and vertical linear profile from 200 m/s at its basis to 250 m/s at its top.

An FTLE advection time of  $T = 0.1$  s was used for the ridge intersection at  $t_0 = 0.4$  s shown in Figure 16(a). Here, aliasing artifacts, a common problem with FTLE ridge extraction in regions of strong separation [4], are obvious. They are most prominent in the upstream part at reverse FTLE ridges. On the other hand, forward FTLE did not produce sufficiently sharp ridges at the downstream end. We excluded both regions during ridge intersection extraction. This left a valid ridge intersection region of roughly 20 m length—located at the center of the domain. By enforcing a threshold of  $\tau_H = -30$  we made sure to extract sufficiently sharp ridges that represent LCS. Starting with an FTLE grid resolution of two times the flow field resolution, two refinement iterations were performed, requiring 98 s of computation time. Finally, short intersection curves were discarded, leaving four ridge intersection curves generating streak manifolds which capture two periods of the flow’s prominent vortex structure (Figure 16(b) and (d)).

As a first test, streak surfaces were generated for  $T_s = 0.1$  s with a seeding offset of 0.15 m, resulting in comparably small streak manifolds (Figure 16(b)). Here, applying reverse preadvection (Section 3.2) prior to streak generation proves particularly useful. A preadvection of the intersection curves by  $T_p = 0.07$  s (Figure 16(c)) enables a total streak generation time of 0.17 s, allowing the streak surfaces to grow larger with the same seeding offset (Figure 16(d)).

The streak manifolds are compared with the reverse FTLE ridges extracted at  $t_0 + T_s = 0.5$  s. The aforementioned wide range in separation strength quickly leads to aliasing of the FTLE ridges, making it difficult to choose the FTLE advection time  $T$  properly to obtain appropriate FTLE ridges for comparison. Here, at the downstream end, an advection time of  $-0.2$  s yields sufficiently sharp reverse FTLE ridges (Figure 16(f)). The ridges were extracted from an FTLE field of four times the flow field resolution. After suppressing insuf-

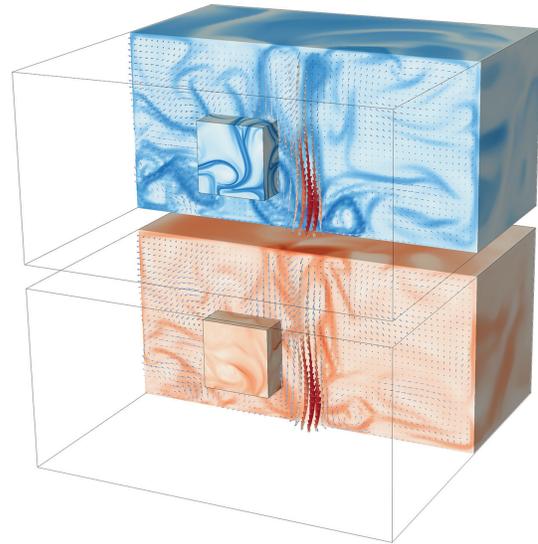


Fig. 12. Buoyant Flow data set overview. Reverse FTLE (top) and forward FTLE (bottom) each with the region of interest at higher resolution and  $T = 20$  s. The vector field behavior is sketched on a plane at  $t = 50.889$  s.

ficiently sharp ridges with the filter  $\tau_H = -20$  for comparison purposes, aliasing is apparent, whereas our streak manifolds are very smooth. Figure 16(g) shows both ridges and streak manifolds. A quantitative analysis of the deviation between streak manifolds and ridges is shown in Figure 16(h): the shortest distance to the ridge surfaces is color-coded on the streak surfaces.

The regions of the streak surface that have an FTLE ridge surface counterpart feature a distance much smaller (blue colors) than the size of a cell (red color) of the FTLE grid used for extracting the ridge surfaces. Note, the accuracy of ridge extraction is in the order of a cell, as indicated in [6]. The color coding reflects the aliasing artifacts of the ridges.

### 5.3 Buoyant Flow

This data set (Section 5.3.1) was computed with a CFD simulation of buoyant air flow inside a closed container. The data set features complex flow behavior and LCS structure allowing us to demonstrate the properties of our approach in cases with many streak manifolds. This includes the influence of hyperbolic strength and the validity of the LCS advection property (Section 5.3.2).

#### 5.3.1 Data Set Description and Overview

The data set has extent  $[0, 10] \times [0, 5] \times [0, 10]$  m<sup>3</sup> on an equidistant grid of resolution  $61 \times 31 \times 61$  for the time interval  $t = [0, 100]$  s. Time discretization is also uniform with  $\Delta t = 1$  s. No-slip adiabatic boundary conditions are applied to all walls. There is a heated plate at the center of the floor and a cooled one at the center of the ceiling. As depicted in Figure 11, another heated plate was added in one of the floor’s corners and a cooled one in the diagonally opposite corner of the ceiling, to increase time-dependent complexity. Figure 12 shows the domain with forward and reverse FTLE

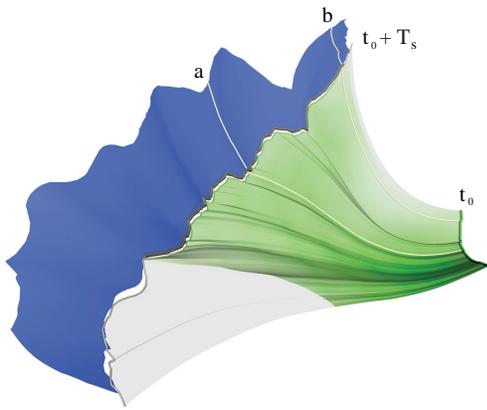


Fig. 13. Hyperbolic path surface from a ridge intersection curve (green, right) of the Buoyant Flow data set ROI. Streak (blue) generation is stopped as non-hyperbolic field regions are entered (gray parts of the path surface). The offset seeding curve at  $t_0 + T_s$  (black) is visualized only in regions that stayed hyperbolic. Comparing streak lines  $a$  and  $b$ , the effect of hyperbolicity strength on streak manifold growth is revealed ( $t_0 = 50.02$  s,  $T_s = 10$  s).

visualization ( $t_0 = 50.889$  s,  $T = 20$  s,  $61 \times 31 \times 61$  FTLE resolution), enhanced with a hedgehog plot overlay at  $t_0$ . In the middle of the domain a strong downward flow is apparent.

Figure 11 shows an overview of the streak manifold topology at time  $t_0 = 50.02$  s with streak generation time  $T_s = 10$  s. A ridge sharpness threshold of  $\tau_H = -3$  was imposed together with a minimum intersection curve length threshold of 0.34 m.

Due to the complex LCS structure we restrict the following investigations to the region of interest from Figure 12, exhibiting strong hyperbolicity responsible for the largest streak manifold. It is located near the center, with an extent of  $[2.5, 5] \times [2, 4.5] \times [4, 5.5]$  m<sup>3</sup>, and a 6 times finer FTLE resolution of  $161 \times 161 \times 121$  for the visualization in Figure 12. Note, half of the ROI is hidden in the overview image.

In the following, the FTLE integration time is set to  $T = 20$  s and  $t_0 = 50.02$  s. For ridge intersection at final level 3 an FTLE threshold of  $\tau_\sigma = 0.15$  was enforced (for obtaining the sharpest ridges) for the reverse FTLE, and 0.16 in the forward one. During refinement a less strict  $\tau_\sigma$  of 0.13 was employed in both fields. We set the seeding offset to 0.02 m and discarded intersection curves shorter than 0.41 m.

### 5.3.2 Hyperbolicity and the LCS Advection Principle

Figure 14 shows an overview of the ROI with the most prominent ridge intersections, and  $T_s = 10$  s. The FTLE at  $t_0$  as well as at  $t_0 + T_s$  is depicted by cross sections. We observe that two nearby reverse FTLE ridges at  $t_0$  “merged” after streak advection time  $T_s$ . At this later instance, the streak manifold is consistent with the attracting FTLE ridge depicted on the section, emphasizing the validity of the advection principle for the attracting LCS. The merging of the less sharp ridge intersections with the sharpest one in the center during streak advection is an example for so-called foliation, i.e., mixing processes captured by LCS.

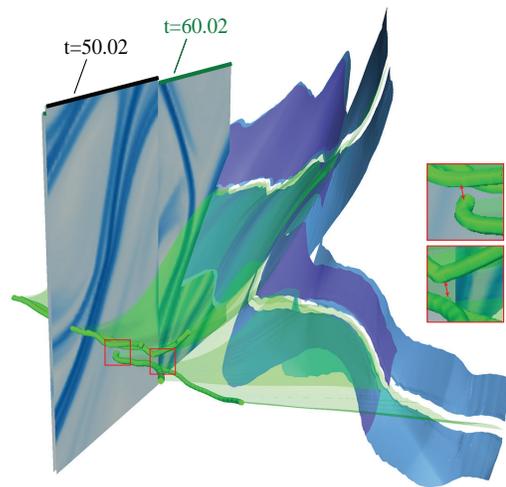


Fig. 14. Streak manifolds of four intersection curves within the ROI of the Buoyant Flow. The corresponding attracting FTLE ridges are depicted on the large cutting plane ( $t = 60.02$  s). The intersection curves are obtained from a single forward FTLE ridge with a “branch” (close-ups) intersecting with two reverse FTLE ridges located closely together (cutting plane,  $t = 50.02$  s). The attracting ridges “merged” at  $t = 60.02$  s ( $t_0 = 50.02$  s,  $T_s = 10$  s).

We additionally investigated the advection principle for the hyperbolic intersection curve. We visualized multiple hyperbolic path surfaces with  $T_s = 7.047$  s emanating from different refinement levels of the ridge intersection curves. Figure 15 shows the ROI with the camera at the bottom pointing upwards, with respect to the camera setting of Figure 14. Again, even at refinement level 0, the advection principle is well satisfied for the attracting LCS, as can be seen at the intersection with the cross section. However, there is a larger discrepancy to the repelling LCS. However, with increasing ridge intersection refinement this discrepancy decreases. Figure 15(b) and (c) additionally visualize hyperbolicity strength mapped to saturation on the hyperbolic path surfaces of level 0 and level 2, respectively. An interesting observation can be made by inspecting hyperbolicity along the path surface: close to the observer it is relatively small, leading to a smaller discrepancy at  $t_0 + T_s$ , whereas near the cross section, the (hyperbolic) effect of the repelling ridge is quite strong, amplifying small errors made in the ridge intersection stage at  $t_0$ . This highlights the importance of our adaptive ridge intersection approach. Please note, however, that deviation from the repelling LCS (or attracting one in the case of reverse streak manifold generation) has usually negligible impact on the accuracy of the resulting streak manifold as long as the streak gets seeded on the correct side of the repelling LCS (Section 5.1.3 and [1]).

Another important aspect of our streak manifold visualization is that the influence of hyperbolicity strength is directly visible in streak growth (Figure 13), enabling a comparative visualization of hyperbolic LCS regions in the ROI (Figure 14) and the whole domain (Figure 11).

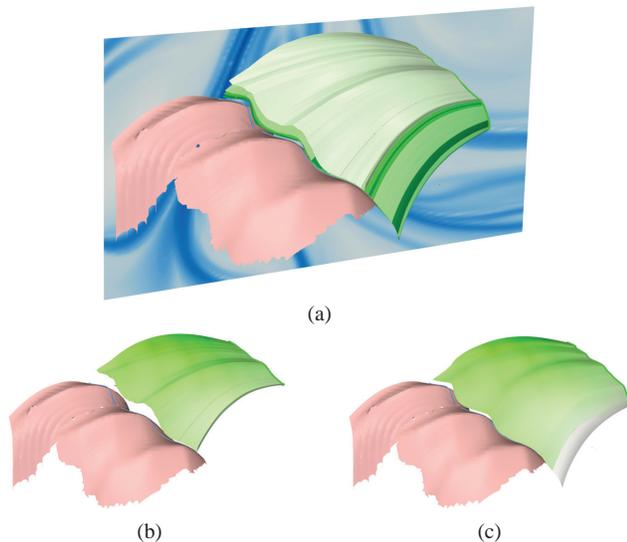


Fig. 15. LCS advection principle. Hyperbolic path surfaces of length  $T_s = 7.047$  s generated by ridge intersection curves at  $t_0 = 50.02$  s obtained at four different refinement levels: Level 0 (light green) to level 3 (dark green). With increasing refinement level of ridge intersection extraction the front of the respective path surface converges toward the ridge intersection at  $t_0 + T_s$ , depicted as blue curve on the forward FTLE ridge (red) (a). Refinement level 0 (b) and level 2 (c) with hyperbolicity color mapping: strongly hyperbolic (green) to non-hyperbolic (white).

## 6 CONCLUSION

We presented the extension of the streak-based time-dependent 2D vector field topology [1] to 3D vector fields. We have shown that in 3D vector fields the counterpart of hyperbolic trajectories are distinguished hyperbolic path surfaces rather than hyperbolic trajectories. These surfaces can be obtained by advecting the intersection curves of attracting and repelling LCS, represented as ridge surfaces in the forward and reverse FTLE fields. We presented an adaptive extraction technique for these intersection curves, providing high accuracy, and efficiency in terms of computation time and memory consumption. Our seeding scheme allows us to generate streak manifolds from the hyperbolic path surfaces. This approach not only gives insight in the space-time structure of time-dependent LCS, but also provides high-quality visualization of LCS compared to the traditional technique by ridge surface extraction from FTLE fields. We evaluated our approach with CFD data and showed that the growth of our streak manifolds additionally visualizes the action of hyperbolic regions and the LCS they cause. As our concept builds upon the integration of streak surfaces, it allows for large time savings when visualizing LCS time series, in comparison to the computationally expensive extraction of LCS by FTLE ridges. We discussed admissible finite-time intervals for inspection by our approach and extend the time interval further by “reverse preadvection”. This allows for longer streak generation phases and hence more significant streak manifolds. We would like to further investigate the properties of streak-based topology, compare it

to other instationary topology concepts, and address LCS that are caused by non-hyperbolic mechanisms, e.g., those induced by shear.

## ACKNOWLEDGMENTS

The authors thank the German Research Foundation (DFG) for support of the project within the *Cluster of Excellence in Simulation Technology* (EXC 310/1) and the Collaborative Research Center SFB-TRR 75 at Universität Stuttgart.

## REFERENCES

- [1] F. Sadlo and D. Weiskopf, “Time-dependent 2-D vector field topology: An approach inspired by Lagrangian coherent structures,” *Computer Graphics Forum*, vol. 29, no. 1, pp. 88–100, 2010.
- [2] G. Haller, “Distinguished material surfaces and coherent structures in three-dimensional fluid flows,” *Physica D*, vol. 149, pp. 248–277, 2001.
- [3] M. Üffinger, F. Sadlo, R. M. Kirby, C. Hansen, and T. Ertl, “FTLE Computation Beyond First Order Approximation,” *Eurographics 2012, Short Papers*, 2012.
- [4] F. Sadlo and R. Peikert, “Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction,” *IEEE Trans. on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 1456–1463, 2007.
- [5] 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*, vol. 212, no. 3-4, pp. 271–304, 2005.
- [6] F. Sadlo, M. Üffinger, T. Ertl, and D. Weiskopf, “On the finite-time scope for computing Lagrangian coherent structures from Lyapunov exponents,” in *Topological Methods in Data Analysis and Visualization II*. Springer, 2012, pp. 269–281.
- [7] T. Wischgoll and G. Scheuermann, “Detection and visualization of closed streamlines in planar flows,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 165–172, 2001.
- [8] J. Helman and L. Hesselink, “Representation and display of vector field topology in fluid flow data sets,” *Computer*, vol. 22, no. 8, pp. 27–36, 1989.
- [9] —, “Visualizing vector field topology in fluid flows,” *Computer Graphics and Applications*, vol. 11, no. 3, pp. 36–46, 1991.
- [10] A. Globus, C. Levit, and T. Lasinski, “A tool for visualizing the topology of three-dimensional vector fields,” in *Proceedings of IEEE Visualization '91*, 1991, pp. 33–41.
- [11] A. Wiebel, X. Tricoche, and G. Scheuermann, “Extraction of separation manifolds using topological structures in flow cross sections,” in *Topology-Based Methods in Vis. II*. Springer, 2009, pp. 31–43.
- [12] X. Tricoche, C. Garth, and G. Scheuermann, “Fast and robust extraction of separation line features,” in *Scientific Visualization: The Visual Extraction of Knowledge from Data*. Springer, 2005, pp. 249–264.
- [13] R. Peikert and M. Roth, “The parallel vectors operator: a vector field visualization primitive,” in *Proceedings of IEEE Visualization '99*, 1999, pp. 263–270.
- [14] D. Kenwright, “Automatic detection of open and closed separation and attachment lines,” in *Proceedings of IEEE Visualization '98*, 1998, pp. 151–158.
- [15] D. Asimov, “Notes on the topology of vector fields and flows,” NASA Ames Research Center, Tech. Rep. RNR-93-003, 1993.
- [16] H. Theisel, T. Weinkauff, H.-C. Hege, and H.-P. Seidel, “Stream line and path line oriented topology for 2D time-dependent vector fields,” in *Proceedings of IEEE Visualization '04*, 2004, pp. 321–328.
- [17] T. Germer, M. Otto, R. Peikert, and H. Theisel, “Lagrangian coherent structures with guaranteed material separation,” *Computer Graphics Forum*, vol. 30, no. 3, pp. 761–770, 2011.
- [18] J. Kasten, I. Hotz, B. Noack, and H.-C. Hege, “On the extraction of long-living features in unsteady fluid flows,” in *Topological Methods in Data Analysis and Visualization*. Springer, 2010, pp. 115–126.
- [19] R. Fuchs, J. Kemmler, B. Schindler, J. Waser, F. Sadlo, H. Hauser, and R. Peikert, “Toward a Lagrangian vector field topology,” *Computer Graphics Forum*, vol. 29, no. 3, pp. 1163–1172, 2010.
- [20] G. Haller, “Finding finite-time invariant manifolds in two-dimensional velocity fields,” *Chaos*, vol. 10, no. 1, pp. 99–108, 2000.
- [21] F. Ferstl, K. Bürger, H. Theisel, and R. Westermann, “Interactive separating streak surfaces,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1569–1577, 2010.

[22] C. Garth, X. Tricoche, T. Salzbrunn, T. Bobach, and G. Scheuermann, "Surface techniques for vortex visualization," in *Proceedings of VisSym*, 2004, pp. 155–164, 346.

[23] C. Garth, H. Krishnan, X. Tricoche, and K. I. Joy, "Generation of accurate integral surfaces in time-dependent vector fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1404–1411, 2008.

[24] H. Krishnan, C. Garth, and K. I. Joy, "Time and streak surfaces for flow visualization in large time-varying data sets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1267–74, 2009.

[25] K. Bürger, F. Ferstl, H. Theisel, and R. Westermann, "Interactive streak surface visualization on the GPU," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1259–1266, 2009.

[26] A. Wiebel, X. Tricoche, D. Schneider, H. Jaenicke, and G. Scheuermann, "Generalized streak lines: Analysis and visualization of boundary induced vortices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1735–1742, 2007.

[27] T. Weinkauff and H. Theisel, "Streak lines as tangent curves of a derived vector field," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1225–1234, 2010.

[28] D. Eberly, *Ridges in Image and Data Analysis*, ser. Computational Imaging and Vision. Kluwer Academic Publishers, 1996.

[29] 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*, vol. 13, no. 6, pp. 1464–71, 2007.

[30] M. Hlawatsch, F. Sadlo, and D. Weiskopf, "Hierarchical line integration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 8, pp. 1148–1163, 2011.

[31] C. Garth and K. I. Joy, "Fast, memory-efficient cell location in unstructured grids for visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1541–1550, 2010.



**Markus Üffinger** received the Diplom (MSc) degree in computer science from the University of Stuttgart, Germany. Since 2007, he has been PhD student at the Visualization Research Center, University of Stuttgart (VISUS). His research interests include scientific visualization and GPU methods.



**Filip Sadlo** received the Diplom (MSc) degree in computer science from ETH Zurich in 2003 where he also did his PhD at the Computer Graphics Laboratory. Since 2008 he has been a research associate at the Visualization Research Center, University of Stuttgart (VISUS). His research interests include scientific visualization, 3D reconstruction, and imaging.



**Thomas Ertl** Thomas Ertl is a full professor of computer science at the University of Stuttgart, Germany and the head of the Visualization and Interactive Systems Institute (VIS) and the Visualization Research Center (VISUS). He received a MS in computer science from the University of Colorado at Boulder and a PhD in theoretical astrophysics from the University of Tübingen. His research interests include visualization, computer graphics, and human computer interaction.

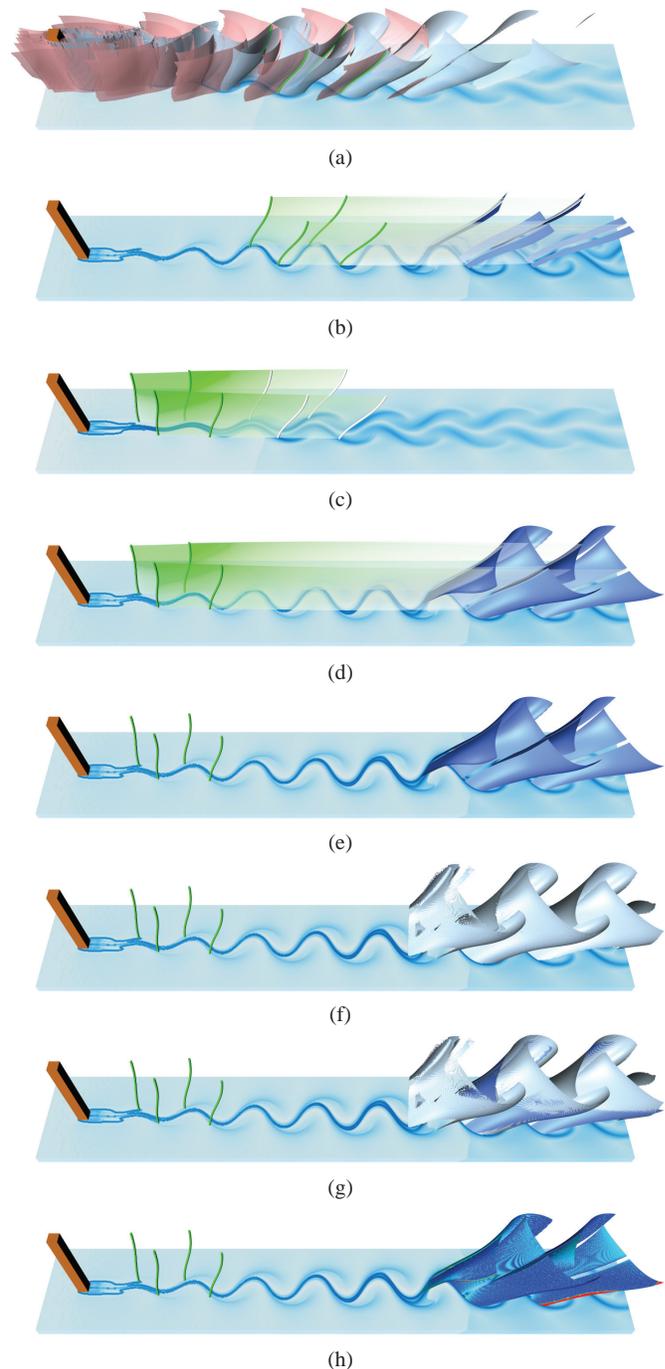


Fig. 16. Von Kármán vortex street example. Ridge intersections (green) at  $t_0 = 0.4$  s (a). Hyperbolic (green) path surfaces and resulting streak manifolds (dark blue) for  $T_s = 0.1$  s, with matching reverse FTLE at  $t = 0.5$  s on bottom plane (b). Reverse preadvective of the intersection curves for  $T_p = 0.07$  s (c) yields larger streak manifolds (d) and (e). Reverse FTLE ridges at  $t = 0.5$  s (f) in comparison to the streak manifolds (g) and with distance mapped to color (h), (zero distance, dark blue). Only regions with no corresponding ridge surface part exhibit distance larger than one FTLE cell (red). The ridges exhibit aliasing, whereas the streak manifolds are of superior quality.