Efficient Computation and Visualization of Coherent Structures in Fluid Flow Applications

Christoph Garth, Florian Gerhardt, Xavier Tricoche, *Member, IEEE* and Hans Hagen, *Member, IEEE*

Abstract—The recently introduced notion of Finite-Time Lyapunov Exponent to characterize Coherent Lagrangian Structures provides a powerful framework for the visualization and analysis of complex technical flows. Its definition is simple and intuitive, and it has a deep theoretical foundation. While the application of this approach seems straightforward in theory, the associated computational cost is essentially prohibitive. Due to the Lagrangian nature of this technique, a huge number of particle paths must be computed to fill the space-time flow domain. In this paper, we propose a novel scheme for the adaptive computation of FTLE fields in two and three dimensions that significantly reduces the number of required particle paths. Furthermore, for three-dimensional flows, we show on several examples that meaningful results can be obtained by restricting the analysis to a well-chosen plane intersecting the flow domain. Finally, we examine some of the visualization aspects of FTLE-based methods and introduce several new variations that help in the analysis of specific aspects of a flow.

Index Terms—flow visualization, feature detection, 3D vector field visualization

1 Introduction

Fluid flows are essential objects of study in a broad range of scientific, engineering, and medical applications. In particular the optimization of numerous industrial processes requires the precise understanding and the control of flows. Examples include important application areas such as combustion, turbomachinery, automotive, and aeronautics. The complexity of the considered flow phenomena and the refined accuracy of the models used for their investigation yield numerical flow datasets whose analysis requires powerful visualization tools to derive the insight necessary for the specific task at hand.

To address the challenge raised by the size and the qualitative complexity of the corresponding vector field datasets, scientific visualization research has explored different approaches that have in common the goal of characterizing, extracting, and visually representing salient flow structures across spatial and temporal scales. These methods are mainly divided into topological and feature-based approaches. While the former leverages a sound mathematical framework and allows for an objective and fully automatic post-processing, the latter explicitly integrates practically significant flow structures into the analysis at the cost of ambiguous definitions and ad-hoc methods. In this context, the notion of Lagrangian Coherent Structures (LCS) and its quantitative assessment using the so-called Finite-Time Lyapunov Exponent (FTLE) provide a promising alternative that combines a well articulated theoretical basis and a physical intuition. Specifically, according to that formalism coherency in steady and transient flows can be characterized in terms of repelling and attracting manifolds. Yet, despite the versatility of this approach and its ability to re-conciliate in a consistent formalism topology and features of interest, its practical application is fundamentally hampered by the prohibitive computational cost associated with the necessary advection of a dense set of particles across the spatio-temporal flow domain.

- *Christoph Garth is with the University of Kaiserslautern, Germany, and the Institute for Data Analysis and Visualization at University of California, Davis, E-mail: cgarth@ucdavis.edu.*
- *Florian Gerhardt is with the University of Kaiserslautern, Germany, E-mail: florian@gerhardtweb.de.*
- *Xavier Tricoche is with the SCI Institute, University of Utah, and Purdue University, E-mail: xmt@cs.purdue.edu.*
- *Hans Hagen is with the University of Kaiserslautern, Germany, E-mail: hagen@informatik.uni-kl.de.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

One of the goals of this work is to reduce this computational cost by significantly constricting the number of particle advections required to perform visualization and analysis based on FTLE and LCS. We propose an incremental, data-driven refinement algorithm that exploits the coherence of particle paths to generate smooth approximations of the so-called flow map, from which the FTLE is computed. The presented approach enables high-resolution analysis of complex 3D flows and permits to construct insightful visualization for accurate assessment of coherence. Furthermore, we show that it is often not necessary to perform the full 3D analysis. Given some a-priori knowledge about the flow field, it is often sufficient and in some cases even beneficial to consider FTLE on planar subsections, further reducing computational effort.

The paper is structured as follows: Section 2 examines past research that is related to our ideas. Since the FTLE and related concepts were only recently introduced in the visualization community, we take the time to review its fundamentals in Section 3, before we present our incremental approximation scheme in Section 4. Section 5 is concerned with discussing possible visualization techniques and introduces the notion of analysis on section planes in 3D flows. Finally, we present results from typical application data in Section 6, and conclude on the presented work in Section 7.

2 Related Work

The use of FTLE as a means to characterize coherent Lagrangian structures in transient flows was introduced by Haller in his seminal paper [6] in 2001. He presented FTLE as a geometric approach, contrasting it with an analytic criterion that he proposed simultaneously. Both approaches aim at characterizing coherent structures in terms of preservation of certain stability types of the velocity gradient along the path of a particle. This work followed previous papers by the same author, investigating similar criteria derived from the eigenvectors of the Jacobian of the flow velocity along pathlines to determine the location of LCS in the two-dimensional setting [5, 9].

This initial research generated a significant interest in FTLE and its applications to the structural analysis of transient flows in the fluid dynamics community, both from a theoretical and from a practical viewpoint. Haller provided a study of the robustness of the coherent structures characterized by FTLE [8] and showed that they are well applicable even under approximation errors in the velocity field. In the same paper, he suggests to identify stable and unstable manifolds with ridge lines of the FTLE field. Shadden et al. provided a more formal discussion of the theory of FTLE and LCS [18] in 2D. One major contribution of their work was to offer an estimate of the flow across the ridge lines of FTLE and to show that it is small and typi-

cally negligible. An extension of FTLE to arbitrary dimensions was discussed in [13]. These tools have been applied to the study of turbulent flows [7, 4, 14] and used in the analysis of vortex ring flows [17]. However, the visualizations presented in these papers were chosen on a case-by-case basis, and no systematic investigation of different visualization types was considered.

Regarding visualization of time-dependent flow fields, several approaches have been explored to permit the extraction and the effective depiction of the structures exhibited by time-dependent flows. Topological methods have been applied to transient flows in the Eulerian perspective [23, 20, 21]. Theisel et al. also proposed a method to characterize the structure of pathlines by subdividing the domain into sink, source, and saddle-like regions based on the divergence of the restriction of the flow to a plane orthogonal to the pathline orientation in space-time [21]. Furthermore, texture-based representations have been considered to visualize time-dependent flows while offering an effective depiction of salient structures, see [12] and references therein. Because of the intrinsic difficulty of defining structures that are both coherent in space and time, each of these methods resorts to some form of ad hoc way to combine the Eulerian and Lagrangian perspectives, leading to animations for which a physical interpretation is typically ambiguous.

Acceleration of FTLE computations for two-dimensional flows was previously achieved by Garth et al. [3] by mapping pathline integration to the GPU. However, their method does not extend to 3D flows due to memory and bandwidth constraints. Sadlo and Peikert [16] also examined the use of FTLE ridge surfaces for visualization of 3D flows. However, they were limited by the resolution that can be achieved in reasonable time with a naive approach.

3 FTLE and Coherent Lagrangian Structures

In the following, we will briefly discuss the basic concepts that are necessary to understand the steps involved in FTLE computations and FTLE-based visualization. Our presentation is voluntarily informal and we refer the interested reader to the publications listed in Section 2 for a more in-depth treatment of this rich subject.

The definition of the FTLE is based on concepts from the theory of dynamical systems. There, the Lyapunov exponent is defined to characterize the rate of separation of infinitesimally close trajectories as time approaches infinity. The idea behind FTLE is to apply this concept in the context of finite-time flow fields and to define asymptotically stable and unstable coherent structures in terms of loci of maximized dispersion of closely seeded particles.

We start by introducing some useful notations. We consider a time-dependent vector field $\mathbf{v}$ defined over a finite Euclidean domain $U \subset \mathrm{R}^3$ and a finite temporal domain $I \subset \mathrm{R}$. The position $\mathbf{x}$ of a particle starting at position $\mathbf{x_0}$ at time t_0 after advection along the vector field can be formulated as a map $\mathbf{x}(t;t_0,\mathbf{x_0})$ satisfying $\mathbf{x}(t_0;t_0,\mathbf{x_0}) = \mathbf{x_0}$ and $\dot{\mathbf{x}}(t;t_0,x_0) = \mathbf{v}(t,\mathbf{x}(t;t_0,x_0))$, where the dot denotes derivative with respect to the first parameter. Specifically, consider a fixed initial time t_0 and a fixed time interval τ, defining $t = t_0 + \tau$. A linearization of the local variation of the map $\mathbf{x}(t,t_0,.)$ around the seed position $\mathbf{x_0}$ is obtained by considering its spatial gradient $J_\mathbf{x}(t,t_0,\mathbf{x_0}) := \nabla_{\mathbf{x_0}}\mathbf{x}(t,t_0,\mathbf{x_0})$ at $\mathbf{x_0}$. This gradient can be used to determine the maximal dispersion after time τ of particles in a neighborhood of $\mathbf{x_0}$ at time t_0 as a function of the direction $\mathbf{d_{t_0}}$ along which we move away from $\mathbf{x_0}$: $\mathbf{d_t} = J_\mathbf{x}(t,t_0,\mathbf{x_0})\ \mathbf{d_{t_0}}$. Maximizing the norm $|\mathbf{d_t}|$ over all possible unit directions $\mathbf{d_{t_0}}$ corresponds to computing the spectral norm of $J_\mathbf{x}(t,t_0,\mathbf{x_0})$. Therefore, maximizing the dispersion of particles around $\mathbf{x_0}$ at t_0 over the space of possible directions around $\mathbf{x_0}$ is equivalent to evaluating

$$\sigma_\tau(t_0,\mathbf{x_0}) := \sqrt{\lambda_{\max}(J_\mathbf{x}(t,t_0,\mathbf{x_0})^T J_\mathbf{x}(t,t_0,\mathbf{x_0}))}.$$

To obtain the average exponential separation rate $\lambda(t,t_0,\mathbf{x_0})$, the logarithm is applied and the result is normalized by advection time τ to obtain

$$\lambda(t,t_0,\mathbf{x_0}) = \frac{1}{|\tau|}\log\ \sqrt{\lambda_{\max}(\ J_\mathbf{x}(t,t_0,\mathbf{x_0})^T J_\mathbf{x}(t,t_0,\mathbf{x_0})\ )}.$$

This rate is then called the *Finite Time Lyapunov Exponent*, and can be evaluated for both forward and backward advection. Large values of λ for forward advection correspond to unstable manifolds while large FTLE values for backward advection correspond to stable manifolds. Shadden et al.[18] showed that ridge lines in these fields correspond to so-called *Lagrangian Coherent Structures*.

In practice, a straightforward approach to computing FTLE fields and LCS is to sample the flow map $\mathbf{x}(t,t_0,.)$ on a regular grid corresponding to the subdomain of interest and approximate its gradient numerically, e.g. by finite differences. The flow map itself is computed by numerical integration in the original vector field. Although this is a conceptually simple procedure, the computational effort in terms of number of pathline integrations makes it almost infeasible even for low sampling resolutions. A typical example with good resolution will require millions of flow map evaluations. In this work, we will consider two independent approaches to reducing the complexity of computation while still maintaining the quality of the resulting visualizations. The following section presents an approach that adaptively approximates the flow map for a given domain with a reduced number of actual pathline integrations. Furthermore, in Section 5.2 we will show that even for 3D flows, restricting the flow map to a planar subsection yields good visualization and analysis results. In combination, these methods allow a significant reduction in computation time w.r.t. the naive approach.

4 Efficient Approximation of FTLE Fields

In the following section we will describe a scheme to adaptively generate a dense sampling of the flow map. The basic premise underlying the following ideas is that evaluation of flow maps is computationally expensive. Using certain convergence properties of subdivision schemes, we have formulated an algorithm for the incremental refinement of flow map approximations that estimates the local approximation quality and refines only where required. This results in a significantly reduced number of pathline integrations.

Before we approach the case of flow maps in 2 or 3 dimensions, we will first introduce the basic principle for the simpler 1d functional case.

4.1 Incremental Approximation of Maps

Assuming a function f on the unit interval, and a discretization of the unit interval with $2^l + 1$ points for $l > 0$, the corresponding discrete samples at the grid points are

$$f_i^l := f\left(\frac{i}{h}\right), \qquad i = 0,\ldots,2^l$$

where $h := \frac{1}{2^l}$. It is readily shown that the piecewise linear interpolant I_f converges point-wise to the function as l increases. More specifically,

$$\lim_{h\to 0} ||f - I_f||_\infty \ \leq\ \max_{[0,1]}(f'')h^2$$

holds (cf. [19]). Therefore, the local convergence speed as $l \to \infty$ allows an estimate of the local variation of f. We next define the subdivision operator S corresponding to linear midpoint subdivision. Its application to f^l yields a new sequence Sf^l with $2^{l+1}+1$ points, i.e.

$$\begin{aligned}(Sf^l)_{2i} &:= f_i^l \\ (Sf^l)_{2i+1} &:= \frac{1}{2}\left(f_i^l + f_{i+1}^l\right)\end{aligned}$$

for $i = 0\ldots 2^l$.

Assuming a discretization f^{l-1} of f exists on level $l-1$, we can construct a *prediction* $\tilde{f}^l$ of the discretization f^l on level l by simply letting

$$\tilde{f}_i^l \ =\ (Sf^{l-1})_i.$$

Comparing $\tilde{f}_i^l$ and f_i^l allows us to estimate the local convergence rate and hence the quality of the approximation. We define

$$d_i^l := ||\tilde{f}_i^l - f_i^l||$$

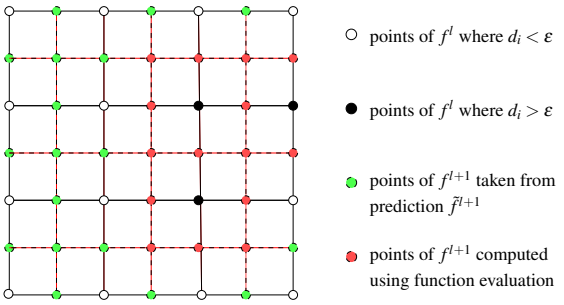


Fig. 1. Incremental refinement form f^l to f^{l+1}.

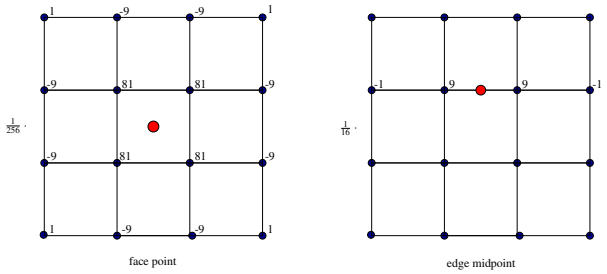


Fig. 2. Subdivision stencils for the tensor-product four-point scheme in 2D. Left: stencil for face points. Right: edge point stencil.

and construct a new approximation on level $l+1$ using

$$f_{2i}^{l+1} := f_i^l$$
$$f_{2i+1}^{l+1} := \begin{cases} \tilde{f}_{2i+1}^{l+1} & \text{if} \max\{d_i^l, d_{i-1}^l\} < \varepsilon \\ f(\frac{2i+1}{2^{l+1}}) & \text{otherwise} \end{cases}$$

In other words, if the local convergence rate is acceptable, we use the predicted value, or replace it with the actual function value otherwise. If ε is chosen small enough and the second derivative of f is bounded almost everywhere, we construct approximation sequences that in the limit converge to the original function in a point-wise sense. For practical purposes, one is usually satisfied with a maximum discretization level $l_{\max}$, and hence irregular behavior of the considered map (such as isolated discontinuities) is unproblematic. The algorithm will simply detect a failure of convergence and refine maximally in these cases.

Using linear subdivision in the above derivation is the simplest possible choice. Unfortunately, the resulting curve is not smooth in the sense that it has piecewise constant derivatives. To obtain approximations of a higher degree of differentiability, the linear subdivision can be replaced by any other type of interpolating subdivision scheme. We have selected the four-point subdivision scheme (cf. [15]) that produces C^1-interpolants in the limit and reproduces cubic polynomials exactly. The corresponding subdivision operator S is given by

$$(Sf^l)_{2i} := f_i^l$$
$$(Sf^l)_{2i+1} := \frac{1}{16}\left(-1f_{i-1}^l + 9f_i^l + 9f_{i+1}^l - 1f_{i+2}^l\right).$$

for $i = 0 \ldots 2^l$. For the boundary points, indices are clamped to 0 and 2^l.

For the functional case, four-point scheme subdivision is equivalent to computing the uniform Catmull-Rom spline interpolating f^l and evaluating it at the interval midpoints to generate the odd points in $\tilde{f}^{l+1}$. Although both descriptions are straightforward in the one-dimensional setting, we feel that extension to higher dimensions using tensor products is much simpler to describe in terms of subdivision. Moreover, the four-point scheme is simply a special case of Kobbelt's $2k$-schemes [11] that produce C^{k-1}-interpolants. It is therefore quite simple to adapt our approach to produce higher degrees of smoothness. Care must be taken, however, in choosing an appropriate k, since higher-order schemes in general tend to overshoot. This is not a conceptual problem for the approximation algorithm, since these overshoots will be corrected during the computation. However, more function evaluations are required.

In this work, the primary application of the presented algorithm is approximation of flow maps, which are then derived to find FTLE fields. Therefore, we have chosen the four-point scheme as a good compromise between required number of evaluations and smoothness mandated by the actual FTLE computation.

4.2 Efficient FTLE Computation

It is quite straightforward to generalize the above derivation to arbitrary dimensions and vector-valued flow maps. To illustrate the adaptive computation, we will describe the refinement algorithm for the 2D case. An identical construction applies to the three-dimensional case as well.

Typically, the 2D domain under consideration is not the unit square but rather an arbitrary rectangle positioned somewhere in the flow domain. To make the following description simpler, we introduce the parameterization P from the unit square to the rectangle and approximate the composition $\mathbf{x}(t, t_0, P(\cdot))$. This has no consequences for the algorithm, but must be kept in mind for later derivative computation. On the same note, the discretization need not have the same size on both axes of the unit square (this makes sense in the case of a strongly elongated rectangle), but for the sake of simplicity we will treat both directions uniformly.

2D refinement algorithm

1. Choose $\varepsilon > 0$, $l > 0$ (typically $l = 4$) and compute the initial flow map approximations f^{l-1} and f^l on the grid with $(2^l+1)^2$ points and spacing $h = \frac{1}{2^l}$. (Note: f^{l-1} is a subset of f^l and therefore entails no additional computations).

2. Compute the distance sequence $d_{i,j}^l$ by point-wise comparison of $f_{i,j}^l$ and $f_{i,j}^{l-1}$

3. Subdivide the approximation f^l to obtain the prediction $\tilde{f}^{l+1}$ on level $l+1$ (Fig. 2).

4. Set $f_{i,j}^{l+1} = \tilde{f}_{i,j}^{l+1}$ if $d_{i',j'} < \varepsilon$ for all grid neighbors that were already present on level l;
   else compute $f_{i,j}^{l+1} = \mathbf{x}(t, t_0, P(\frac{i}{2^{l+1}}, \frac{j}{2^{l+1}}))$. (cf. Fig. 1)

5. If maximum level not reached, continue at 2.

The end result of this algorithm is a dense flow map approximation that can then be used for visualization and analysis purposes. Figure 3 illustrates the adaptivity of our algorithm by showing the distribution of flow map evaluations for a specific example. The ε parameter indirectly determines the accuracy of refinement. It directly refers to the distance between predicted and actual flow map result and thus depends on the scale of the dataset under consideration and the contained structures therein and must be chosen adequately. While this might seem problematic at first glance, our experiments indicate that this does not seem an issue in practice.

An implementation of the incremental refinement scheme is quite straightforward. Assuming that pathline integration and dataset management are already provided for (e.g. by an external toolkit), one has to essentially provide code for n-dimensional array management and the four-point subdivision scheme. Our model C++ implementation consists of about 250 lines of code.

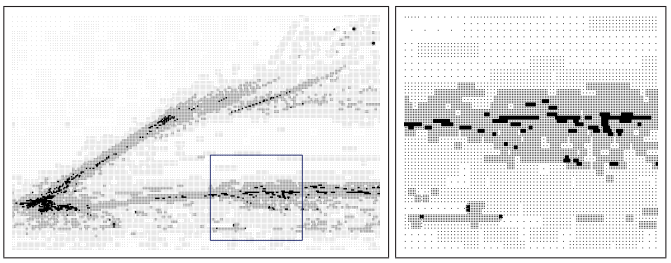


Fig. 3. Visualization of the incremental refinement algorithm on the plane from Fig. 6: black points indicate evaluations of the flow map. The right image shows a close-up of the framed region.

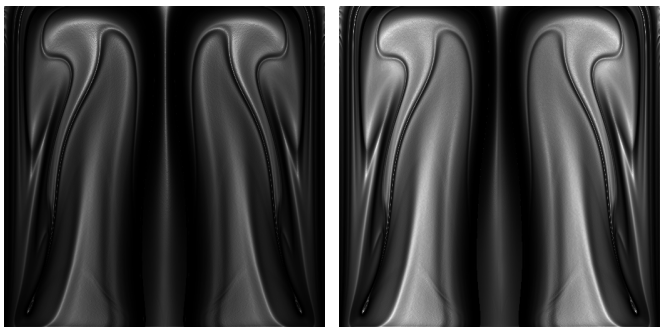


Fig. 5. Comparison of planar FTLE vs. a section of 3D FTLE in the cylinder dataset (dark areas correspond to high values of forward FTLE). The chosen section is perpendicular to the lid and contains the cylinder axis.

5 Visualization of Coherent Flow Structures

Because of its objective and fully automatic nature, the data-driven FTLE computation described in the previous section can be carried out in an offline manner. This yields a high-resolution, two- or three-dimensional (potentially unsteady) scalar field that quantifies the structural coherence of the flow. We now describe how the resulting information can be leveraged from a visualization standpoint to produce images that effectively support the insightful exploration of complex flow structures in practically relevant datasets. We organize our presentation with respect to the dimension of the considered FTLE field, which allows us to underscore the specific features associated with each setting.

5.1 Three-dimensional FTLE

FTLE volume data constitute a direct and conceptually straightforward means to characterize the structural contents of a selected subregion of the flow domain. In essence, two complementary approaches can be used to visualize this information. The first one corresponds to displaying the whole three-dimensional FTLE data in a way that naturally underscores the fuzzy nature of individual coherent structures. This type of representation is powerful since it offers an overview of the flow that emphasizes its most prominent features. As such, it also provides a context for more targeted visualizations. In contrast, the second approach derives surfaces from the FTLE volume to create a skeleton-type representation that reduces the visual complexity of the visualization and explicitly computes a structural segmentation of the domain.

Volume rendering To transform the FTLE volume data into a volume rendering visualization we map forward-time and backward-time FTLE to the axes of a two-dimensional transfer function domain. As demonstrated in previous publications [22], this approach permits to identify interesting correlations between the different dimensions of the transfer function space and therefore allows for a very selective definition of features of interest. We use this idea to capture the shape of significant flow patterns and isolate them for display.

Ridge surfaces The natural way to derive a surface-type visualization for FTLE volume data consists of extracting the corresponding ridge surfaces. In a general setting these surfaces correspond to manifolds along which the scalar field is locally maximized. This definition extends to arbitrary dimensions the intuitive notions of ridges and valleys in a (two-dimensional) height field. More specifically, the definition of these surfaces involves gradient and Hessian of the considered scalar quantity. In our case, with λ denoting the FTLE field corresponding to either forward or backward advection, we can characterize ridge surfaces as the set of points satisfying following conditions (cf. [2]):

$$\mathbf{g}_\lambda . \mathbf{e}_{min} = 0, \text{ with } \mu_{min} < 0, \tag{1}$$

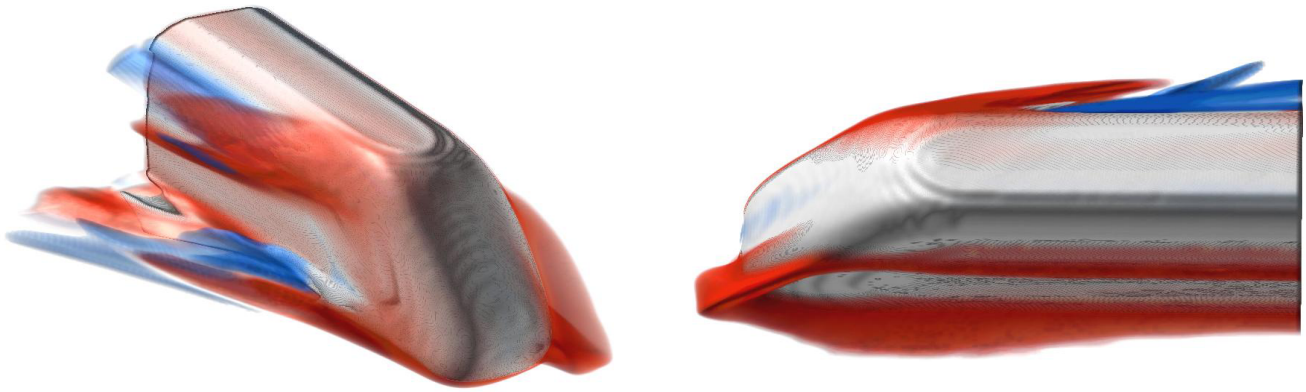
where $\mathbf{g}_\lambda = \nabla\lambda$ designates the (spatial) gradient of λ, $\mu_{min} \leq \mu_{medium} \leq \mu_{max}$ are the ascending eigenvalues of λ's Hessian $\mathbf{H}_\lambda = \nabla^2\lambda$ (which is a 3x3 matrix in the three-dimensional case that we consider here), and $\mathbf{e}_{min}$ is the eigenvector associated with μ_{min}. A natural way to extract these ridge surfaces consists of applying an approach similar to *Marching Triangles* to identify 0-surfaces of $\mathbf{g}_\lambda . \mathbf{e}_{min}$ (see [10]). This requires to assign an arbitrary but locally consistent orientation to the eigenvector field $\mathbf{e}_{min}$. While the corresponding implementation seems straightforward, the intrinsically noisy nature of the FTLE field combined with the fact that ridge surface extraction involves its first and second order derivatives make its reliable computation remarkably challenging. It is worthwhile to note that the point coordinates that enter the FTLE computation are inevitably noisy due to the inaccuracies of the numerical integration used to compute them across unstructured CFD grids. We have experimented with the method proposed in [10] to extract such surfaces from noisy test datasets but the results we obtained were not satisfactory given the lack of smoothness of the FTLE field and the structural complexity of our test datasets. It is worthwhile to note that the results obtained with our datasets strongly contrast with those previously achieved with less complex flow data that makes ridge surface extraction a more feasible task [16].

5.2 FTLE on Planar Cross Sections

A FTLE computation on a cross section provides a means to characterize the coherence of the particles that intersect the corresponding region. Compared to the three-dimensional case it also reduces the number of particles that must be advected and thus the overall complexity of the analysis. While in its original form, the definition of the FTLE is based on the 3×3 Jacobian J_x of the flow map, a planar sampling can only provide two spatial derivatives, resulting in a 3×2-Jacobian. The definition of the spectral norm, however, carries across to these non-square matrices, such that the FTLE derivation can be applied to planar samplings of the flow map as well.

It is important to point out that, despite its lack of formal justification, this approach is based on the following reasoning. Going back to its definition (Section 3), FTLE is designed to measure the separation of neighboring trajectories under the action of an underlying unstable hyperbolic manifold. It has been shown in [17] that Lagrangian Coherent Structures (LCS) play the role of material lines and therefore no exponential separation can occur along them in general. Therefore, the eigenvector of the flow map Jacobian associated with the maximal separation is roughly perpendicular to the local flow direction. Hence, we hypothesize that a valid approximation of FTLE can be achieved by restricting the particle sampling to a plane that is mostly orthogonal to the flow direction (and by extension to suspected LCS). Of course, if the plane is not exactly orthogonal to the interesting structure, the stability of the eigenvalue computation is affected. However, the maximal separation rate of neighboring trajectories seeded on the plane will be captured qualitatively nevertheless, as indicated by our experiments (see also Section 6 and Fig. 5).

Additionally, the analysis applied to the resulting scalar image offers an effective basis to perform the complex, time-consuming, and typically error-prone task of seeding integral curves and surfaces to



(a) Right side of front wagon: multiple structures are visible. Of notable interest are the attachment structure at the rear end of the wagon and the corresponding attachment at middle height. They hint at vortex shedding at the top edge of the wagon.

(b) Two attachment structures on the left side of the train, originating at the nose. The upper structure allows identification of the flow going over the top of the wagon.

Fig. 4. ICE train: Volume rendering of forward (red) and backward (blue) FTLE fields.

probe a three- or four-dimensional flow domain. In the course of our experimentation we have investigated several techniques using this basic idea that we describe next.

Ridge lines The definition introduced for the 3D case applies similarly in the 2D case to characterize ridge lines. The numerical challenges associated with their extraction are somewhat mitigated by the lower dimension even if it remains difficult. In a recent paper, Mathur et al. [14] proposed an iterative scheme combining an integration along the gradient and a stop criterion based on Equation 1. In contrast the solution that we adopted in our work uses smooth and analytic reconstruction kernels for the derivatives of the FTLE field similar to [10] and applies filtering criteria based on feature length and ridge strength (as measured by μ_{min}) to discard weak features and false positives. The resulting lines can then be used to seed a dense set of neighboring particles or a stream surface.

Stochastic particle seeding An FTLE field may be interpreted as a Probability Density Function (PDF) for particle seed distribution. The rejection rule applied to this data is then used to determine stochastically a set of seed points that overall emphasize regions of high FTLE values and as such of high structural coherence. The fuzziness of the resulting representation matches naturally the fuzziness of flow coherence and the uncertainty involved in their computational characterization.

Image-based user interface Planar FTLE images provide the user with a look-up map over which interesting regions that may be difficult or impossible to extract automatically can be manually and selectively identified by simple brushing to provide a PDF similar to the one mentioned previously but this time geared towards the specific focus of the analysis. This reduces the visual complexity of the final image, to emphasize most prominent aspects in the data, and it provides an intuitive interface to do so.

6 Results

To study the performance and viability of the presented visualization methods, we have applied them to a large number of application datasets, three of which we will present in detail in this section.

6.1 Visualization Results

6.1.1 Rotating Lid Cylinder

The instationary simulation describes a can filled with a highly viscous fluid that is accelerated by rotation of the lower lid of the can. Due to the high viscosity of the fluid and the high degree of symmetry the velocity field is of very good numerical quality. Although the dataset is symmetric w.r.t. the rotation axis, the simulation was computed on the full configuration. The grid is fixed for all 5000 time steps and consists of 750.000 unstructured cells.

We have used this dataset mainly for benchmark purposes since it does not contain any flow regions without structures. Furthermore, we have used it to study the differences between planar FTLE and a slice of the three-dimensional FTLE field (cf. Section 5.2). Fig. 5 shows a direct comparison. While the 3D FTLE slice provides more contrast, the observable structures are qualitatively identical.

6.1.2 ICE train

This dataset is the result of a stationary simulation of a high-speed train traveling at at velocity of 250 km/h with wind blowing form the side at an angle of 30 degrees. Empirically, the wind causes vortices to form on the leeward side of the train. The resulting drop of pressure on this side in combination with the increased pressure on the windward side create a pressure differential that adversely affects the train's track holding. The computational grid is medium sized with 2.6 million unstructured elements in total.

Fig. 4 shows a Direct Volume Rendering of the FTLE structures

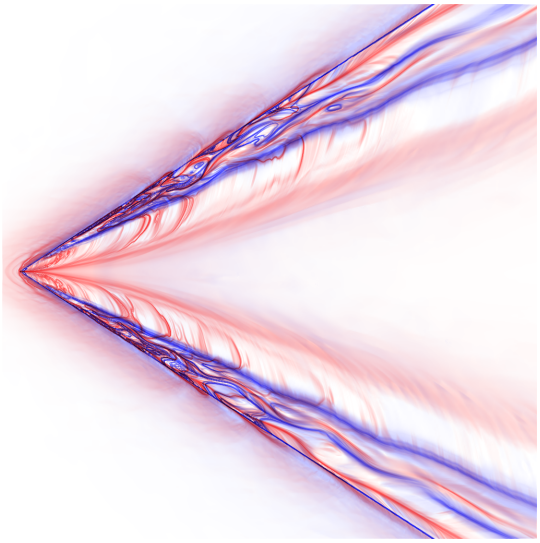
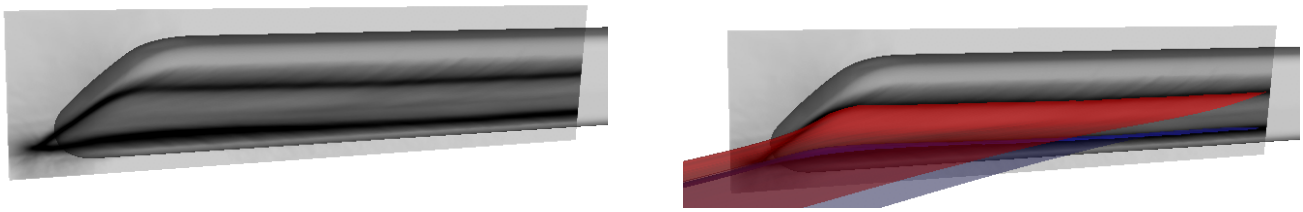


Fig. 7. Closeup of the TDELTA wing tip: planar FTLE reveals highly non-trivial separation (red) and attachment (blue) structures and demonstrates the complexity of flow structures in modern CFD datasets.



(a) Separation strength (forward FTLE).

(b) Stream surfaces integrated from FTLE ridge lines.

Fig. 6. ICE train: Left side separation structures are extracted using a cutting-plane FTLE analysis with a plane parallel and close to the side of the train. Extraction of forward FTLE ridges delivers starting curves for stream surface integration.

Dataset	**time-dep.**	**Resolution**	Evaluations	rel. l_2-error	rel. l_∞-error
TDELTA wing	no	1024×1153	12%	$5.32 \cdot 10^{-5}$	$7.38 \cdot 10^{-3}$
TDELTA slice	no	2048×1024	7%	$6.64 \cdot 10^{-5}$	$9.18 \cdot 10^{-3}$
Cylinder	*yes*	2048^2	9%	$6.79 \cdot 10^{-8}$	$9.12 \cdot 10^{-6}$
Can dataset	*yes*	128^3	14%	$9.43 \cdot 10^{-7}$	$1.33 \cdot 10^{-1}$
ICE train	*no*	$289^2 \times 65$	5%	$3.81 \cdot 10^{-4}$	$3.37 \cdot 10^{-1}$
TDELTA box	*no*	$257 \times 321 \times 65$	25%	$9.59 \cdot 10^{-6}$	$2.6 \cdot 10^{-3}$

Table 1. Incremental approximation test results.

in forward and backward time in a box around the nose of the front wagon. The images permit a good overview of the prevalent separation and attachment structures close to the surface. On the right side, vortex shedding at the top edge of the wagon can be inferred. On the left side, the separation structures show that a significant volume of air streams above the top of the wagon and contributes to the vortex shedding on the right side. To obtain a more detailed depiction of these separation lines, we have computed a high-resolution planar FTLE section along and close to the left side of the train (Fig. 6(a)). From this section, we extracted ridge lines of forward FTLE and used them as starting curves for stream surfaces, see Fig. 6(b). Using this approach, it was possible to quickly achieve a detailed picture of these separation structures.

6.1.3 Delta wing

The so-called TDELTA dataset focuses on the study of the transient flow above a delta wing at low speeds and increasing angle of attack. The main goal was to numerically investigate the cause of vortex breakdown that the primary vortices exhibit. Of special interest is the connection between attachment and separation structures on the wing surface and the vortices above the wing. The instationary simulation features a time-varying adaptive grid. Unfortunately, due to its enormous size, only three (non-successive) time steps were available to us. Therefore, we are limited to stationary considerations here. The selected timestep grid consists of about 15 million unstructured elements.

From a visualization point of view, the main challenge in this dataset is the complexity of the flow above the wing with its nesting and interacting coherent structures. Fig. 8 provides an overview. The outermost layers occlude most of the inner structures, therefore, auxiliary techniques such as clipping planes or volume cropping have to be applied to obtain a good depiction of the flow. Overall, however, the insight provided by 3D FTLE visualizations is insufficient. To reduce visual complexity, we have again applied the planar FTLE technique. Fig. 9(a) shows very short streamlines stochastically seeded in regions of high forward and backward planar FTLE directly above the wing, giving a very good illustration of alternating surface separation and attachment lines. The separation surfaces emanating from these lines are further examined using planar FTLE on an plane perpendicular to the wing axis (cf. Fig. 9(b)). The resulting 2D image (Fig. 9(c) shows the impressive complexity of the flow above the wing. Several vortices are clearly visible, and the separation surface emanating from the wing edge exhibits a rolling-up type structure. To generate 3D visualizations from this image, we have employed user-driven fuzzy pathline seeding (see painted areas in Fig. 9(c) and corresponding streamlines in Fig. 9(b)). This allows a further study of the observed structures. For example, we observed a point of very high FTLE in the center of the right primary vortex (yellow dot). The corresponding streamlines show the typical vortex breakdown bubble strength and explain this point as the strong separation generated by the upstream swirl saddle that usually accompanies the breakdown bubble.

6.2 Performance Analysis

To study the performance of adaptive approximation, we have examined a number of test cases. For each of these, we have compared the result of our approximation algorithm against a flow map sampling obtained by brute force. In Table 1, the corresponding results are detailed concerning the percentage of flow map evaluations as compared to the full sampling, and the resulting mean deviation (relative l_2-error) and maximum deviation (relative l_∞-error).

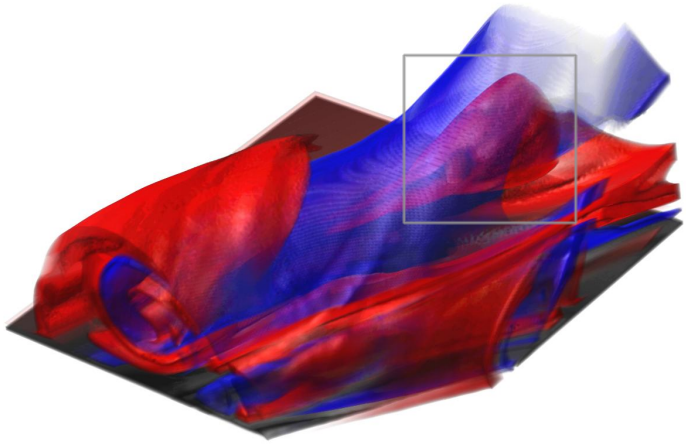
Overall, the incremental refinement approach allows a reduction of computational effort by a factor of 4 to 10. We have found that in the initial exploration of a dataset, the region of interest is chosen such that it contains a large volume with respect to typical structure size. This is mostly because the location of these structures is not a-prior known. Hence, this is the case that profits most from the incremental approximation. As the region of interest is refined to the scale of the actual structures, the benefits are reduced. We therefore believe that our algorithm is ideally suited to the exploration of datasets, where it can provide significantly reduced computation times.

Furthermore, our numerical experiments show that as the parameter ε decreases, the relative errors with respect to the non-adaptive computation tend towards zero. Therefore, we are confident that our ideas are applicable to datasets of very high complexity.

7 Conclusion

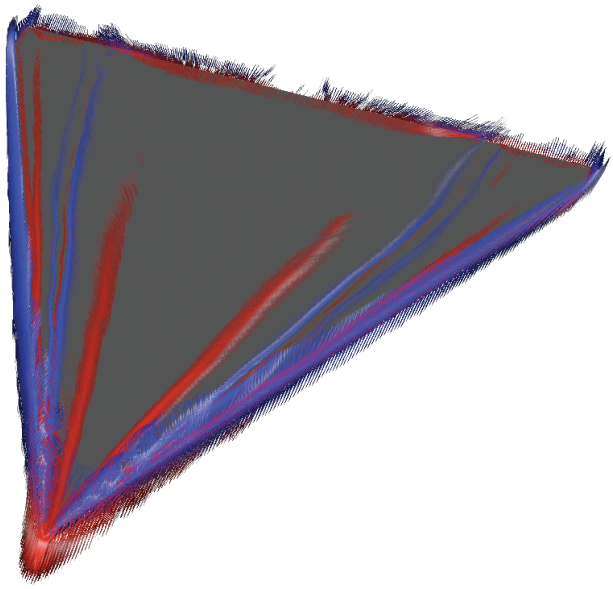
In this paper, we have presented an incremental algorithm for the computation of flow map approximations and FTLE fields that allows for a significant reduction of computational effort, allowing for high-resolution visualization and analysis of FTLE and the corresponding Lagrangian Coherent Structures. In all our experiments, our algorithm was stable and delivered good performance improvements. Moreover, we have demonstrated the power of FTLE-based visualization methods on several examples from relevant application areas. Furthermore, we have shown that even for three-dimensional flows, planar FTLE analysis and derived methods such as particle seeding yield insightful

(a) Wing edge separation and the primary attachment layer. Inner structures are occluded.

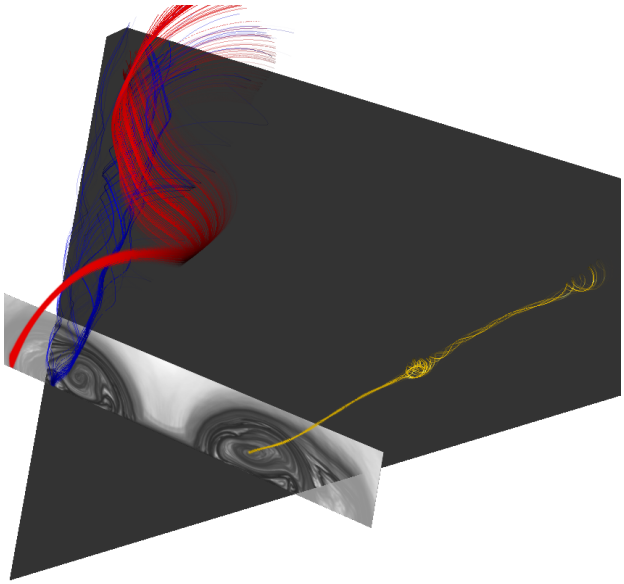


(b) Crop along the middle third of the left wing edge. The interplay of separation and attachment structures is visible on the front face. The grey box highlights the separation structure that characterizes a vortex breakdown bubble

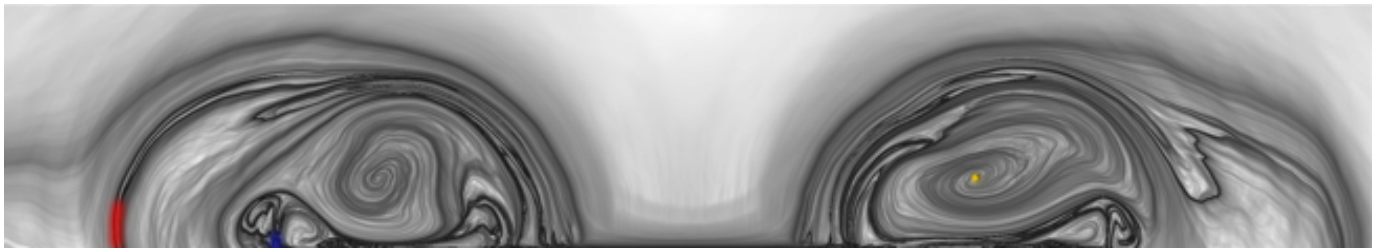
Fig. 8. TDELTA Dataset: Volume Rendering of Regions of high forward (red) and backward (blue) FTLE. Coherent Structures appear as surfaces. Occlusion is problematic and has to be resolved through the use of cropping or clipping.



(a) Separation and attachment structures on the wing surface. Pathlines were seeded on the forward and backward ridges of planar FTLE on a directly above the wing.



(b) Visualization of primary (red) and secondary (blue) separation structures. Pathlines seeded according to PDF in (c).



(c) Planar FTLE visualization on a section plane perpendicular to the main flow direction. Darker regions correspond to regions of high FTLE. Colored regions indicate PDFs that were used to seed pathlines for (b).

Fig. 9. TDELTA Dataset: Fuzzy pathline visualization of flow structures above the wing. Pathlines are seeded according to FTLE strength, FTLE ridge lines or through a user-guided Probability Density Function.

results and can further reduce computation times and facilitate user-guided exploration of datasets.

There are many avenues for future work:

- In this paper, we have only considered application of the incremental refinement algorithm in 2D and 3D settings. However, it should be possible to compute time-varying 3D FTLE fields directly using this approach.

- In the current form, our algorithm does not take into account derived information such as e.g. ridges to steer the refinement. However, this seems worth considering in the future.

- Computation of the flow map is in essence a highly parallel task, since the pathline computations are mutually independent. It is therefore possible to exploit parallel machine architectures to further accelerate these computations.

- While we have provided some effective visualization types, we would like to devise coherent structure visualization that effectively combine particle, line, surface, and volume representations, possibly in combination with illustrative rendering methods, to convey the progressive transitions between neighboring or intertwined flow structures. On the same note, we would like to examine new user interfaces that facilitate working with FTLE maps and derived visualization types.

- In this paper, we have not discussed the influence of integration length on the FTLE computations. Also, quite recently, a number of flow-derived quantities closely related to FTLE, such as e.g. the Finite Separation Lyapunov Exponent [1], have appeared in the literature. These aspects should be systematically examined.

ACKNOWLEDGEMENTS

The authors wish to thank Markus Rütten from DLR Göttingen for supplying the datasets treated here and insightful discussion. We would also wish to extend our gratitude to Gordon Kindlmann and Joe Kniss for making great software available. Both *teem* and *Simian* were very useful in the development of the presented methods. We are also very much indebted to the Visualization Group at the University of Leipzig for ongoing discussion and collaboration. During the course of this work, Christoph Garth was financially supported by the DFG International Research Training Group "Visualization of Large and Unstructured Datasets" based at the University of Kaiserslautern.

REFERENCES

[1] E. Aurell, G. Boffetta, A. Crisanti, G. Paladin, and A. Vulpiani. Predictability in the large: An extension of the concept of lyapunov exponent. *J. Phys. A: Math. Gen*, 30:1 – 26, 1997.

[2] D. Eberly, R. Gardner, B. Morse, and S. Pizer. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4:355–371, 1994.

[3] C. Garth, G.-S. Li, X. Tricoche, C. D. Hansen, and H. Hagen. Visualization of coherent structures in 2d transient flows. In *Topology-Based Methods in Visualization, Proceedings of the 2007 Workshop*, page to appear, 2007.

[4] M. Green, C. Rowley, and G. Haller. Detection of lagrangian coherent structures in 3d turbulence. *J. Fluid Mech.*, to appear, 2006.

[5] G. Haller. Finding finite-time invariant manifolds in two-dimensional velocity fields. *Chaos*, 10(1):99–108, 2000.

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

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

[8] G. Haller. Lagrangian coherent structures from approximate velocity data. *Physics of Fluids*, 14(6):1851–1861, june 2002.

[9] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D*, 147:352–370, 2000.

[10] G. Kindlmann, X. Tricoche, and C.-F. Westin. Anisotropy creases delineate white matter structure in diffusion tensor mri. In *Proceedings of Medical Imaging Computing and Computer-Assisted Intervention, MICCAI '06*, 2006.

[11] L. Kobbelt. *Iterative Erzeugung glatter Interpolanten*. Ph.D. Thesis, Univ. Karlsruhe, 1994.

[12] B. Laramee, J. van Wijk, B. Jobard, and H. Hauser. ISA and IBFVS: Image space based visualization of flow on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):637–648, nov 2004.

[13] F. Lekien, S. Shadden, and J. Marsden. Lagrangian coherent structures in n-dimensional systems. *Physica D*, submitted, 2006.

[14] M. Mathur, G. Haller, T. Peacock, J. Ruppert-Felsot, and H. Swinney. Uncovering the lagrangian skeleton of turbulence. *Phys. Rev. Lett.*, submitted, 2006.

[15] H. Prautzsch, W. Boehm, and M. Paluszny. *Bézier and B-Spline Techniques*. Springer Berlin, 2002.

[16] F. Sadlo and R. Peikert. Visualizing lagrangian coherent structured and comparison to vector field topology. In *Topology-Based Methods in Visualization, Proceedings of the 2007 Workshop*, page to appear, 2007.

[17] S. Shadden, J. Dabiri, and J. Marsden. Lagrangian analysis of fluid transport in empirical vortex ring flows. *Physics of Fluids*, 18:047105, 2006.

[18] S. Shadden, F. Lekien, and J. Marsden. Definition and properties of lagrangian coherent structures from finit-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212:271–304, 2005.

[19] W. G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Cambrige Press, 1973.

[20] H. Theisel and H.-P. Seidel. Feature flow fields. In *Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '03)*, pages 141–148, 2003.

[21] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Topological methods for 2d time-dependent vector fields based on streamlines and path lines. *IEEE Transations on Visualization and Computer Graphics*, 11(4):383–394, 2005.

[22] X. Tricoche, C. Garth, G. Kindlmann, E. Deines, G. Scheuermann, M. Rütten, and C. Hansen. Visualization of intricate flow structures for vortex breakdown analysis. In *Proceeding of IEEE Visualization '04 Conference*, pages 187–194, 2004.

[23] X. Tricoche, T. Wischgoll, G. Scheuermannn, and H. Hagen. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computer and Graphics*, 26:249–257, 2002.