Topological Methods for 2D Time-Dependent Vector Fields Based on Stream Lines and Path Lines

Holger Theisel, Tino Weinkauf, Hans-Christian Hege, and Hans-Peter Seidel

(Invited Paper)

Abstract— **This paper describes approaches to topologically segmenting 2D time-dependent vector fields. For this class of vector fields, two important classes of lines exist: stream lines and path lines. Because of this, two segmentations are possible: either concerning the behavior of stream lines, or of path lines. While topological features based on stream lines are well established, we introduce path line oriented topology as a new visualization approach in this paper. As a contribution to stream line oriented topology we introduce new methods to detect global bifurcations like saddle connections and cyclic fold bifurcations as well as a method to tracking all isolated closed stream lines. To get the path line oriented topology we segment the vector field into areas of attracting, repelling and saddle-like behavior of the path lines. We compare both kinds of topologies and apply them to a number of test data sets.**

Index Terms— **flow visualization, vector field topology, bifurcations, stream lines, path lines**

I. Introduction

TOPOLOGICAL methods have become a standard tool in vector field visualization. Initially introduced as a visualization tool in [1], topological methods have been extended to higher order critical points [2], boundary switch points [3], and closed separatrices [4]. In addition, topological methods have been applied to simplify [3] [5] [6] [7], smooth [8], compress [9] and design [10] vector fields. The topology of 3D vector fields is visualized in [11], [12], [13], [14], [15].

For 2D time-dependent vector fields there exists a number of extensions of topological concepts. [16] and [17] track the location of critical points over time and detect local bifurcations like fold bifurcations and Hopf bifurcations. This approach works on a piecewise linear vector field and computes and connects the critical points on the faces of a prism cell structure, which is constructed from the underlying triangular grid. [4] introduce a method to detect closed stream lines in 2D steady vector fields. This method also relies on the underlying triangular grid of the piecewise linear vector field. [18] extends this to 2D time-dependent vector fields by applying an approach similar to contouring and connecting of isosurfaces: closed stream lines are extracted for every time step, then corresponding lines in adjacent time steps are connected. As every contouring and connecting approach, special attention has to be paid to events like appearance, disappearance or collapsing of closed stream lines. [19] introduces a method to track critical points in a time-dependent vector field which does not depend on a particular underling grid: the paths of the critical points are tracked as the stream lines of a new vector field called feature flow field which can be extracted from the original vector field.

The main motivation behind topological methods is to segment a vector field into areas of similar flow behavior which is determined by observing the behavior of certain characteristic curves. For 2D time-dependent vector fields, two important classes of curves exist: stream lines and path lines. Hence, two different kinds of topologies can be considered: a stream line oriented topology where areas are segmented which show a similar behavior of stream lines, and a path line oriented topology which does so for path lines. All above-mentioned 2D time-dependent topological methods are stream line oriented.

This paper describes an extended version of [20]. Based on the distinction of stream line and path line oriented topology, we make two major contributions: for stream line oriented methods, we propose new approaches to detect global bifurcations like saddle connections and cyclic fold bifurcations. In addition we propose a new approach to detect and track closed stream lines. This approach does not depend on an underlying grid structure, and it does not have to solve the correspondence problem between adjacent time steps neither. The second major contribution is the consideration of path line oriented topology. This kind of topology has not been considered in the visualization community yet. For this, we divide the vector field into areas where the path lines show attracting, repelling, or saddle-like behavior respectively.

The rest of the paper is organized as follows: section

H. Theisel and H.-P. Seidel are with the MPI Informatik Saarbrücken, E-mail: {theisel, hpseidel}@mpi-sb.mpg.de

T. Weinkauf and H.-C. Hege are with the Zuse Institute Berlin, E-mail: {weinkauf, hege}@zib.de

II recalls the concepts of stream lines and path lines and gives a setup to distinguish and analyze them. Section III treats the streamline oriented topology: after reviewing the most important concepts and previous work (section III-A), we introduce a new method to detect saddle connections (section III-B). Section III-C presents a new method to track closed stream lines which is independent of an underlying grid and robust against cyclic fold bifurcations. To apply this, we need an appropriate system of starting closed stream lines which is introduced in section III-D. Section IV presents the approach to a path line oriented topology by segmenting the vector field into areas of different path line behavior. In section V we apply both concepts of topology to a number of test data sets. Conclusions are drawn in section VI.

II. Stream Lines and Path Lines

We are interested in spatio-temporal characteristics of a time-dependent vector field vector field $\mathbf{v}(\mathbf{x},t)$ defined in some space-time domain D. In a space-time point $(\mathbf{x}_0,t_0) \in D$ we can start e.g. a *path line*

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{v}(\mathbf{x}(t),t) \quad \text{with} \quad \mathbf{x}(t_0) = \mathbf{x}_0 \qquad (1)$$

which may be written equivalently in integral form

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{v}(\mathbf{x}(s),s)\,ds \qquad (2)$$

or a *stream line*, staying in some time slice $t = t_0$,

$$\frac{d}{d\tau}\mathbf{x}(\tau) = \mathbf{v}(\mathbf{x}(\tau),t_0) \quad \text{with} \quad \mathbf{x}(0) = \mathbf{x}_0 \qquad (3)$$

which reads in integral form

$$\mathbf{x}(\tau) = \mathbf{x}_0 + \int_{0}^{\tau} \mathbf{v}(\mathbf{x}(s),t_0)\,ds. \qquad (4)$$

The ODE system (1) can be rewritten as an autonomous system at the expense of an increase in dimension by one, if time is included as an explicit state variable:

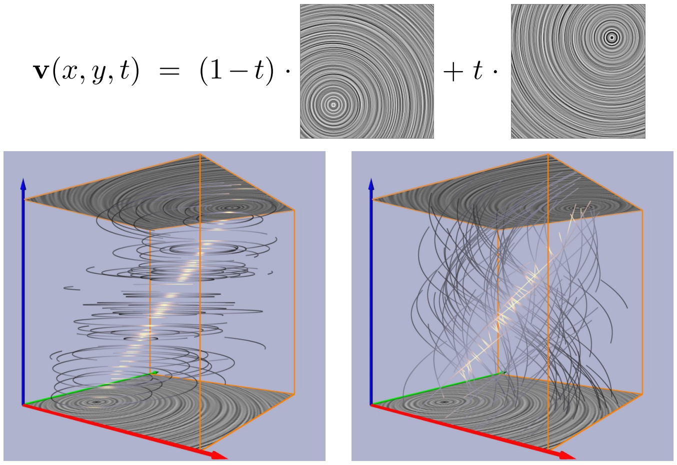
$$\frac{d}{dt}\begin{pmatrix}\mathbf{x}\\ t\end{pmatrix} = \begin{pmatrix}\mathbf{v}(\mathbf{x}(t),t)\\ 1\end{pmatrix} \text{ with } \begin{pmatrix}\mathbf{x}\\ t\end{pmatrix}(0) = \begin{pmatrix}\mathbf{x}_0\\ t_0\end{pmatrix}. \qquad (5)$$

In this formulation space and time are dealt with on equal footing – facilitating the analysis of spatio-temporal features. Path lines of the original vector field $\mathbf{v}$ in ordinary space now appear as stream lines of the vector field

$$\mathbf{p}(x,y,t) = \begin{pmatrix}\mathbf{v}(\mathbf{x},t)\\ 1\end{pmatrix} \qquad (6)$$

in space-time. To treat streamlines of $\mathbf{v}$, one may simply use

$$\mathbf{s}(x,y,t) = \begin{pmatrix}\mathbf{v}(\mathbf{x},t)\\ 0\end{pmatrix}. \qquad (7)$$



(a) Stream lines of $\mathbf{s}$ correspond to the stream lines in $\mathbf{v}$.

(b) Stream lines of $\mathbf{p}$ correspond to the path lines in $\mathbf{v}$.

Fig. 1. Characteristic curves of a simple 2D time-dependent vector field shown as illuminated field lines.

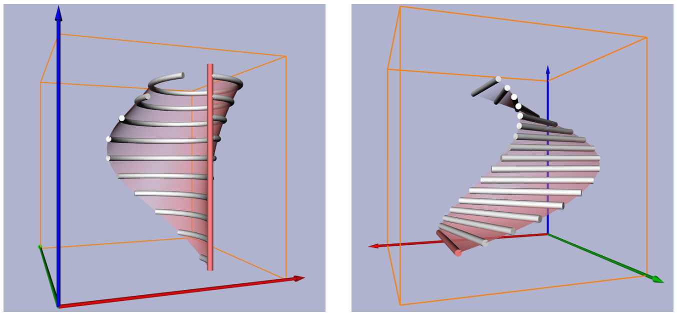
This is valid for arbitrary space dimensions. In the following we restrict our considerations to a 2D time-dependent vector field

$$\mathbf{v}(x,y,t) = \begin{pmatrix}u(x,y,t)\\ v(x,y,t)\end{pmatrix}. \qquad (8)$$

Figure 1 illustrates $\mathbf{s}$ and $\mathbf{p}$ for a simple example vector field $\mathbf{v}$. It is obtained by a linear interpolation over time of two bilinear vector fields. Note that in all figures throughout this paper the coordinate system is shown as follows: red/green coordinate axes denote the (x,y)-domain, the blue axis shows the time component.

Now the problem of finding a stream line and path line oriented topology is simply reduced to finding the topological skeletons of $\mathbf{s}$ and $\mathbf{p}$. Unfortunately, neither for $\mathbf{s}$ nor for $\mathbf{p}$ the classical vector field topology extraction techniques for 3D vector fields are applicable: $\mathbf{s}$ consists of critical lines while $\mathbf{p}$ does not have any critical points at all. Sections III and IV show how to treat their topological skeletons.

For 2D time-dependent vector fields there are also other classes of characteristic curves, namely streak lines and time lines. A streak line is the location of all particles set out at different times but the same location. Considering the vector field $\mathbf{p}$ introduced above, streak lines can be obtained in the following way: apply a stream surface integration in $\mathbf{p}$ where the seeding curve is a straight line segment parallel to the t-axis, a streak line is the intersection of this stream surface with a plane perpendicular to the t-axis (Figure 2a). A time line can be obtained by applying a stream surface integration in $\mathbf{p}$ starting at a line with $t =$ const., and intersecting it with a plane perpendicular to the t-axis (Figure 2b).



(a) Streak lines of $\mathbf{v}$. (b) Time lines of $\mathbf{v}$.

Fig. 2. Characteristic curves of a simple 2D time-dependent vector field. Seeding curves and resulting stream surfaces are colored red. Same data set as in Figure 1.

Both streak and time lines fail to have a property of stream and path lines respectively: they are not locally unique, i.e. for a particular location and time there is more than one streak and time line passing through. Hence they cannot be described as stream lines of a certain 3D vector field. Because of this we restrict ourselves to stream line and path line oriented topology here.

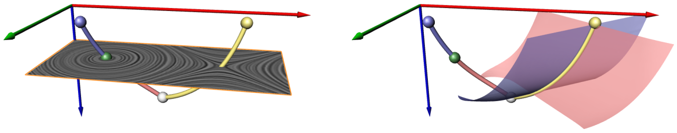
III. Stream Line Oriented Topology

Stream line oriented topology is well-understood in the visualization community ([21], [22], [23]). In addition to tracking the topological features over time, bifurcations have to be extracted. Bifurcations are the events of structural changes of the flow behavior at a certain time. We first review the most important concepts and approaches for their visualization (section III-A) before we introduce new methods to detect certain global bifurcations: saddle connections (III-B) and cyclic fold bifurcations (III-C).

A. *Concepts and previous work*

Critical points are important topological features of steady vector fields. Tracking their location over time is necessary for capturing the topological behavior of $\mathbf{v}$. This is equivalent to extracting the zero lines of $\mathbf{s}$, where all points on these lines are zero points of $\mathbf{v}$ at a certain time. To do so, one can either extract and connect the zeros on the faces of an underlying prism cell grid ([16]), or a feature flow field integration from a start zero point of $\mathbf{s}$ is applied. The feature flow field for tracking critical points is a 3D vector field $\mathbf{f}$ which points into the direction where all components of $\mathbf{s}$ remain unchanged. [19] shows that

$$\mathbf{f}(x,y,t) = \begin{pmatrix} \det(\mathbf{v}_y, \mathbf{v}_t) \\ \det(\mathbf{v}_t, \mathbf{v}_x) \\ \det(\mathbf{v}_x, \mathbf{v}_y) \end{pmatrix}. \tag{9}$$



(a) Critical lines of $\mathbf{s}$, LIC plane through Hopf bifurcation. (b) Separation surfaces created by the moving saddle.

Fig. 3. Topological visualization of a simple 2D time-dependent vector field consisting of sink, source, saddle, fold and Hopf bifurcation – one of each type.

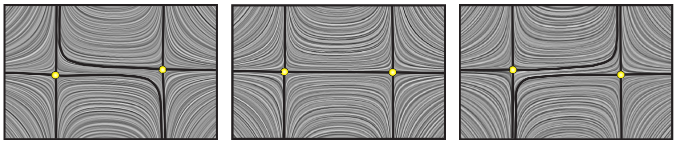
Starting a stream line integration of $\mathbf{f}$ from a point $\mathbf{x}_0$ with $\mathbf{s}(\mathbf{x}_0) = (0,0,0)^T$, all points $\mathbf{x}$ on this stream line fulfill $\mathbf{s}(\mathbf{x}) = (0,0,0)^T$ as well. Here we prefer the feature flow field approach to extract the critical lines of $\mathbf{s}$ since it does not depend on an underlying grid.

To extract all critical lines of $\mathbf{s}$, an appropriate number of start points is needed. We get them by considering all critical points at the boundaries of the domain of $\mathbf{s}$ (which can easily be obtained as critical points of 2D vector fields) and by additionally considering all *fold bifurcations* of $\mathbf{v}$. A fold bifurcation appears if at a certain time t a critical point appears and in the same moment splits up to a saddle and source/sink/center.[1] Fold bifurcations can be found as the zeros of the following system of equations: $[u = 0,\ v = 0,\ \det(\mathbf{v}_x, \mathbf{v}_y) = 0]$. This refers to local extrema of the t-values on the critical lines. To solve this system for isolated (x,y,t), we use a simple subdivision approach similar to detecting isolated critical points in 3D vector fields: a cell C in the domain is checked whether one of the components (u, v, or $\det(\mathbf{v}_x, \mathbf{v}_y)$) is positive at all 8 vertices of C (or whether one component is negative at all vertices). If so, no fold bifurcation is found in C. Otherwise we recursively subdivide C into 8 subcells until their size is smaller than a certain threshold. The results are clusters of small cells which represent the isolated fold bifurcations.

Another important class of local bifurcations are *Hopf bifurcations* denoting locations where a sink becomes a source or vice versa. Thus, this denotes the location of a center, i.e. a critical point with a vanishing divergence and a positive Jacobian. Hopf bifurcations can be extracted similar to fold bifurcations by numerically solving the system $[u = 0,\ v = 0,\ \mathrm{div}(\mathbf{v}) = u_x + v_y = 0]$ for (x,y,t) and selecting all isolated solutions with positive Jacobian.

Another part of the topological skeleton of $\mathbf{v}$ are the separation curves starting from saddle points. It is a well-known fact that a saddle of a 2D vector field creates 4 separation curves by starting the integration

[1] Or the other way around: a saddle and a source/sink/center collapse and disappear.



(a) Shortly before. (b) The event. (c) Shortly after.

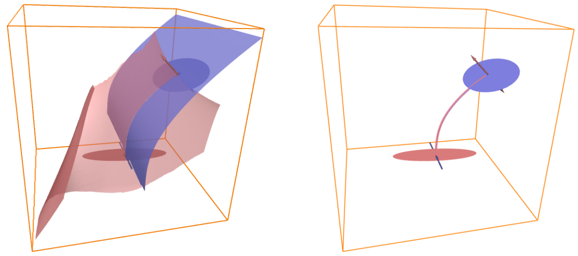
Fig. 4. Saddle connection bifurcation.

into the directions of the eigenvectors of the Jacobian matrix. While the saddle moves over time in $\mathbf{v}$, their sweepings form 4 stream surfaces dividing $\mathbf{s}$ into areas of different flow behavior. Figure 3 gives an illustration of a simple vector field containing all topological features mentioned above. In this figure (as well as in the following figures) we use the following color coding: the critical lines of $\mathbf{s}$ are color coded according to the inflow/outflow behavior of the represented critical points in $\mathbf{v}$: a red/blue/green/yellow line segment represents a source/sink/center/saddle critical point respectively. The same color coding is used for particular critical points which are visualized as small spheres. This means that a Hopf bifurcation is shown as a small green sphere. Furthermore, fold bifurcations are shown as gray spheres, while particular stream lines of $\mathbf{s}$ are shown as gray lines. For integrated separation surfaces we color code according to the integration direction as red (forward integration) or blue (backward integration) surfaces.

B. Detecting saddle connections

Saddle connections are global bifurcations which appear when two separatrices starting from saddle points coincide, i.e. when a separatrix of one saddle ends in another saddle. Figure 4 illustrates an example. We are not aware of pre-existing solutions to extracting all saddle connections of $\mathbf{v}$ for visualization purposes.

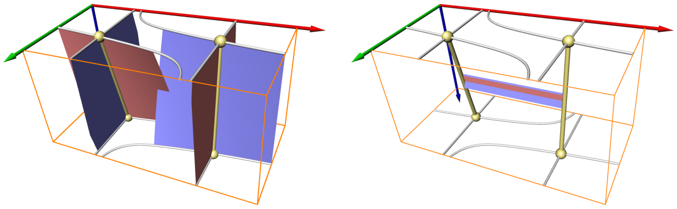
The solution we propose here is an adaption of the saddle connectors approach [14] which recently has been introduced to visualize topological skeletons of 3D steady vector fields.[2] Saddle connectors are the intersection curves of the separation surfaces of a 3D vector field starting in the outflow and inflow planes of the saddle points. The basic idea is to numerically integrate two separation surfaces until an intersection is found. After some refinement, a stream line is integrated from the intersection point both in forward and backward direction. Figure 5 illustrates the idea of saddle connectors, details about implementation, accuracy and speed are in [14].



(a) Separation surfaces of the saddles.

(b) Their intersection curve is the saddle connector.

Fig. 5. Definition of saddle connectors for 3D vector fields.



(a) Separation surfaces starting from critical lines of $\mathbf{s}$.

(b) Saddle connection as the intersection of these surfaces.

Fig. 6. Extracting saddle connections.

Now the idea of saddle connectors can be modified to detect saddle connections in $\mathbf{s}$: instead of starting the integration of one separation surface at each saddle of a 3D vector field, we start in the critical lines of $\mathbf{s}$ which represent a moving saddle. In fact, we start four stream surface integrations[3] at the critical lines of $\mathbf{s}$ into the directions of the eigenvectors of the Jacobian matrix. The rest of the algorithm is similar to saddle connectors and yields all saddle connections in $\mathbf{v}$. Figure 6 gives an illustration.

A special case of saddle connections is the so-called *periodic blue sky bifurcation* ([22]) where two separatrices of the same saddle coincide. Our algorithm to extract saddle connections automatically extracts these bifurcations as well. Figure 7 illustrates this.

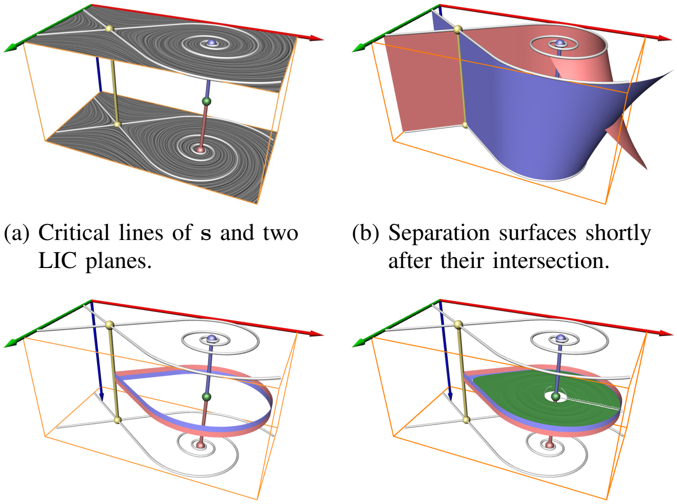
C. Tracking closed streamlines

Closed stream lines are global topological features which evolve over time in $\mathbf{v}$. Several bifurcations can occur: a closed stream line may appear or disappear, or two closed stream lines may collapse and disappear. The latter case is called *cyclic fold bifurcation* and is illustrated in figures 8 and 10.

A first approach to track closed stream lines was proposed in [16]: closed stream lines are extracted in different time levels, and corresponding stream lines in

[2] Note the nice coincidence of concepts: saddle connections (of 2D time-dependent vector fields) are extracted using an adaption of saddle connectors (for 3D steady vector fields).

[3] Two forwards and two backwards.



(a) Critical lines of $\mathbf{s}$ and two LIC planes.

(b) Separation surfaces shortly after their intersection.

(c) Two separation curves of the same saddle coincide; the intersection with the yellow line is the location of the saddle creating the bifurcation.

(d) Tracked closed stream line starting from Hopf bifurcation.

Fig. 7. Periodic blue sky bifurcation.

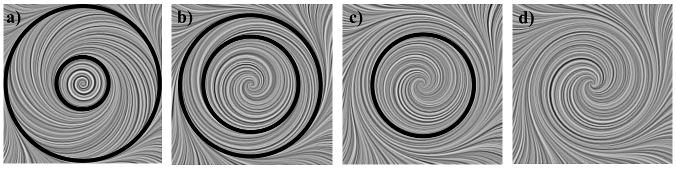
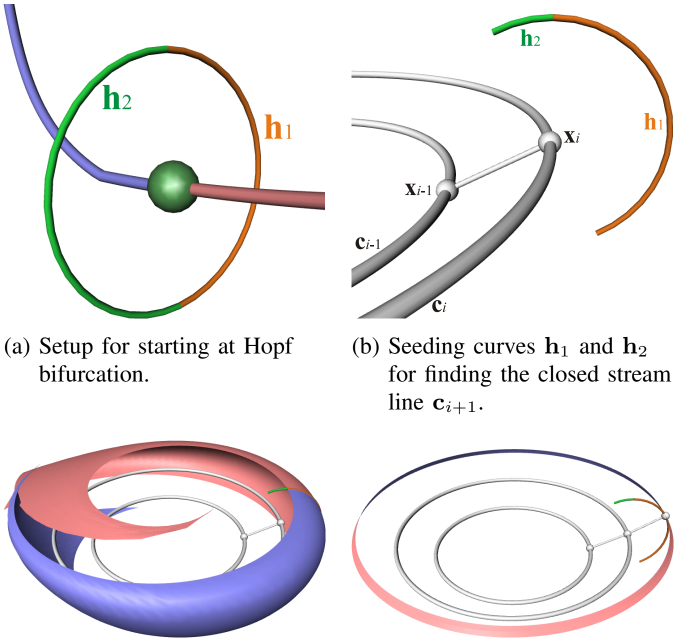


Fig. 8. Cyclic fold bifurcation: two closed stream lines move towards each other (a and b), merge (c) and disappear (d).

adjacent time levels are connected. The results are tube-shaped surfaces starting/ending in Hopf bifurcations, periodic blue sky bifurcations, or at the boundaries of the domain. This approach depends on the underlying grid structure and does not consider cyclic fold bifurcations.

We present a new solution for tracking closed stream lines without these restrictions. This section describes the primary part of the algorithm by assuming a starting closed stream line is already given. How to obtain such a starting line is explained in the next section.

Our new approach follows the general idea of feature flow fields [19]. Suppose we already have a closed stream line $\mathbf{c}_i$. We would like to construct a 3D vector field $\mathbf{g}$ such that the evolution of $\mathbf{c}_i$ over time can simply be obtained by a stream surface integration of $\mathbf{g}$ starting at $\mathbf{c}_i$. Unfortunately, such a feature flow field $\mathbf{g}$ cannot locally be derived from $\mathbf{s}$, since the evolution of a closed stream line is a global process. But the basic principle of feature flow fields can be reduced to this: given $\mathbf{c}_i$, we want to find an adjacent closed stream line $\mathbf{c}_{i+1}$ within a certain distance d from $\mathbf{c}_i$. Furthermore, d shall refer to the (x, y, t)-domain, i.e. we do not make any assumption whether the t-value of $\mathbf{c}_{i+1}$ is before, after, or at the same t-value as $\mathbf{c}_i$. Note that a (closed) stream line is uniquely defined by a single point on it. Hence, we only have to construct a point $\mathbf{x}_{i+1}$ on $\mathbf{c}_{i+1}$ with a certain distance from a point $\mathbf{x}_i$ on $\mathbf{c}_i$. To achieve this, we again apply an adaption of the concept of saddle connectors [14].



(a) Setup for starting at Hopf bifurcation.

(b) Seeding curves $\mathbf{h}_1$ and $\mathbf{h}_2$ for finding the closed stream line $\mathbf{c}_{i+1}$.

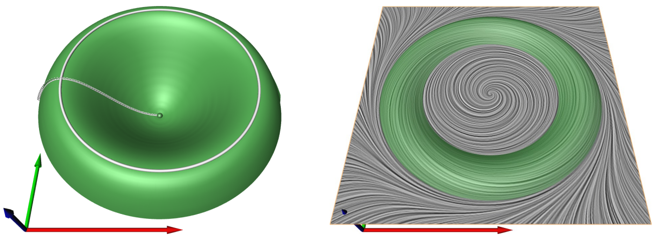
(c) Intersecting stream surfaces evolving from $\mathbf{h}_1$.

(d) Detected connector is $\mathbf{c}_{i+1}$.

Fig. 9. Tracking closed stream lines.

We describe one step of our algorithm now (Figures 9b-d). Given two adjacent closed stream lines $\mathbf{c}_{i-1}$ and $\mathbf{c}_i$ together with their defining points $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ (Figure 9b), we want to find a point $\mathbf{x}_{i+1}$ which defines the next closed stream line $\mathbf{c}_{i+1}$. To do so, we consider a plane through $\mathbf{x}_i$ perpendicular to $\mathbf{s}(\mathbf{x}_i)$ and place a circle $\mathbf{k}$ around $\mathbf{x}_i$ with the radius d into this plane: every point on $\mathbf{k}$ represents a certain step in space-time. It is easy to see, that if $\mathbf{c}_{i+1}$ is actually existing, then $\mathbf{x}_{i+1} \in \mathbf{k}$ is fulfilled. Since we assume the closed stream line to evolve continuously, we search $\mathbf{x}_{i+1}$ only on a circular arc $\hat{\mathbf{k}} \subset \mathbf{k}$ consisting of all points $\mathbf{x} \in \mathbf{k}$ with $(\mathbf{x} - \mathbf{x}_i)(\mathbf{x}_i - \mathbf{x}_{i-1}) > 0$ (Figure 9b). This ensures, that the algorithm does not run back to $\mathbf{c}_{i-1}$. Anyway, if $\mathbf{c}_{i-1}$ is not given we have $\hat{\mathbf{k}} = \mathbf{k}$. $\hat{\mathbf{k}}$ is further subdivided by cutting along the t-direction into two arcs $\mathbf{h}_1$ and $\mathbf{h}_2$.[4] They act as seeding curves of a stream surface integration of $\mathbf{s}$ in both forward and backward direction (Figure 9c).

[4] The splitting of $\hat{\mathbf{k}}$ into $\mathbf{h}_1$ and $\mathbf{h}_2$ is necessary to ensure that each found intersection is a closed stream line. If we would start the integration on $\hat{\mathbf{k}}$, the intersection curves could be stream lines starting and ending in different points, i.e. not closed stream lines, since two points on $\hat{\mathbf{k}}$ may have the same t-value.



(a) Stream surface describing closed stream lines. Cyclic fold bifurcation is detected on top of this surface (gray line).

(b) Intersection of stream surface with LIC plane. Compare with Figure 8b.

Fig. 10. Detecting a cyclic fold bifurcation.

Similar to the saddle connector approach, we detect the intersection of these stream surfaces: it describes a closed stream line (Figure 9d).

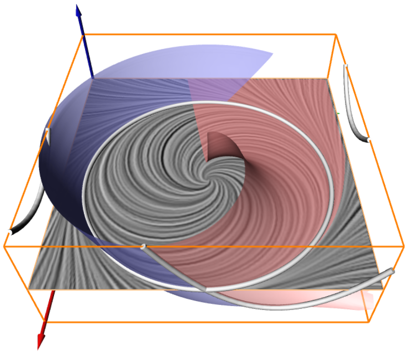
Assuming length$(\mathbf{h}_1) \geq$ length$(\mathbf{h}_2)$, we first apply the stream surface integration starting in $\mathbf{h}_1$. This is encouraged by the fact that the direction of the closed stream line's evolution in this step is unlikely to differ much from the previous direction. Our experience shows that this yields much faster execution times. If an intersection curve is found, this is the new closed stream line $\mathbf{c}_{i+1}$; its intersection with $\mathbf{h}_1$ gives $\mathbf{x}_{i+1}$. If no intersection is found, we check $\mathbf{h}_2$ in a similar way. If this gives no result either, the algorithm stops.

After extracting a sequence of closed stream lines, there are two ways for extracting the stream surface describing them. One is to use the curve $[\mathbf{x}_0, \ldots, \mathbf{x}_n]$ as a seeding line for another stream surface integration. However, since we already extracted a number of closed stream lines, we use them to triangulate the strips between each adjacent $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$. The result is a triangular mesh representing a stream surface describing the evolution of a closed stream line in space-time.

Figure 10 shows an example of a tracked closed stream line starting at a Hopf bifurcation. We see the sequence of defining points $\mathbf{x}_i$ as well as the searching arcs $\mathbf{h}_1$ and $\mathbf{h}_2$ in every step. This example shows that the algorithm can deal with cyclic fold bifurcations: it appears at the closed stream line $\mathbf{c}_i$ if $(t_i - t_{i-1})(t_{i+1} - t_i) \leq 0$ (where t_i is the t-component of the points on $\mathbf{c}_i$). The example in Figure 10 has one cyclic fold bifurcation (gray line).

Another example is shown in figure 7d, where a Hopf and a periodic blue sky bifurcation have the same t-value. It turns out that the closed stream line completely evolves in the same time-slice. This is an example where tracking approaches fail which are based on extracting and connecting closed stream lines in fixed time steps.

The numerical integration of the seeding line $[\mathbf{x}_0, \ldots, \mathbf{x}_n]$ has the advantage that numerical errors are not inherited: if due to numerical errors $\mathbf{x}_i$ is located a little bit away from a closed stream line, $\mathbf{x}_{i+1}$ can still be found correctly.



11. A closed stream line touching the boundary of the domain is obtained by intersecting stream surfaces of $\mathbf{s}$ starting from the boundary switch curves.

D. *Initial conditions for tracking closed streamlines*

To complete the algorithm of tracking closed stream lines, we still have to find a system of initial closed stream lines, i.e. closed stream lines where the tracking described in section III-C starts or ends. This system of initial stream lines has to be chosen in such a way that all closed stream lines are guaranteed to be tracked. To do so, we choose all events where a closed stream line appears or disappears. These events are:

- Hopf bifurcations.
- Periodic blue sky bifurcations.
- Closed stream lines at the first or last time step.
- Closed stream lines touching the boundary of the domain and appearing/disappearing subsequently.

At a Hopf bifurcation $\mathbf{x}_0$, we start the integration in a semi-circle around $\mathbf{x}_0$ which lies in the plane defined by $(0,0,1)^T$ and $(0,0,1)^T \times \mathbf{f}(\mathbf{x}_0)$, where $\mathbf{f}$ is the feature flow field for tracking critical points (9). Figure 9a illustrates this. At a periodic blue sky bifurcation, we take any point on the stream line as starting point $\mathbf{x}_0$. Setting $\hat{\mathbf{k}}$ as the full circle $\mathbf{k}$, the rest of the integration step is as described above.

To detect closed stream lines at the first and last time step of $\mathbf{v}$ means to detect all closed stream lines in 2D steady vector fields. [24] shows that this problem is analogous to intersecting certain stream surfaces in a 3D vector field. Based on this, [24] presents an algorithm to detecting all closed stream lines in a 2D steady vector field.

To find closed stream lines touching the boundary of the domain of $\mathbf{v}$, we extract the boundary switch curves ([15]) of $\mathbf{s}$. From these lines we start a stream surface integration of $\mathbf{s}$ both in backward and forward direction. Their intersections indicate a closed stream line touching the boundary of the domain of $\mathbf{s}$. Figure 11 gives an illustration.

IV. Path Line Oriented Topology

Constructing a path line oriented topology means to consider the stream lines of $\mathbf{p}$ (cf. equation (6)) and segment $\mathbf{p}$ into regions of different flow behavior. Although an extensive research about extracting characteristic structures of path lines ([25], [26]) and local fluid particle motion ([27]) has been done in the fluid dynamics community, topological approaches of path lines have not been used as a visualization tool yet.

A. *Definition of path line oriented topology*

Given a 3D vector field $\mathbf{w}$, it is a well-known fact that a number of local properties (like acceleration, shear, curvature, convergence and others) can be computed by a local analysis of the Jacobian matrix $\mathbf{J_w}$ ([27]). The usual approach to compute them is a decomposition of $\mathbf{J_w}$ together with introducing a local coordinate system. These properties also have been used in icon-based visualization tools for 3D vector fields [28]. The local properties of $\mathbf{w}$ can be divided into two groups: the first group consist of pure directional properties of the stream lines of $\mathbf{w}$, i.e. they are invariant under scalings of $\mathbf{w}$. The second group consists of properties which depend both on the length and the direction of $\mathbf{w}$.

The basic approach we use here is to exploit local properties of a 3D vector field to get a segmentation of $\mathbf{p}$. Since a topological skeleton should only depend on the direction of the stream lines, we focus here on properties of the first group. In fact, we aim at segmenting $\mathbf{p}$ into areas of attracting, repelling, or saddle like behavior of the stream lines of $\mathbf{p}$.

Consider a point $\mathbf{x}_0$ with its corresponding vector $\mathbf{p}(\mathbf{x}_0)$ defining a stream line ℓ_0 in $\mathbf{p}$ starting at $\mathbf{x}_0$. We consider a small circle $\mathbf{k}_0$ around $\mathbf{x}_0$ in the plane perpendicular to $\mathbf{p}(\mathbf{x}_0)$. Considering all stream lines ℓ in $\mathbf{p}$ starting in $\mathbf{k}_0$, three stable cases are possible concerning their convergence/divergence behavior towards ℓ_0:

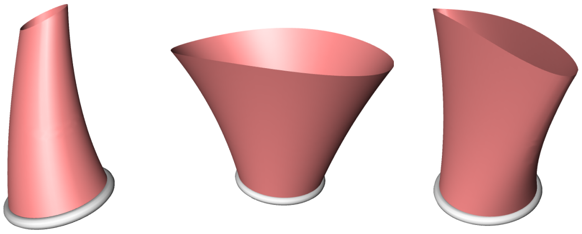
- All ℓ converge towards ℓ_0 under forward integration.
- All ℓ move away from ℓ_0 under forward integration.
- Stream lines with both converging and diverging behavior exist (saddle-like behavior).

Figure 12 illustrates these cases.

To compute this classification as a local property of $\mathbf{x}_0$, we transform $\mathbf{p}$ into a local coordinate system such that the new origin is $\mathbf{x}_0$ and the new base vectors $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are

$$\mathbf{b}_3 = \frac{\mathbf{p}(\mathbf{x}_0)}{\|\mathbf{p}(\mathbf{x}_0)\|}\ ,\ \mathbf{b}_1 = \frac{(0,-1,v)^T}{\sqrt{1+v^2}}\ ,\ \mathbf{b}_2 = \mathbf{b}_3 \times \mathbf{b}_1. \quad (10)$$

Note that $\mathbf{b}_1$ is chosen to be non-zero and perpendicular to $\mathbf{b}_3$. Any other choice of $\mathbf{b}_1$ fulfilling these demands is possible as well. Since we are only interested in local properties perpendicular to the flow direction, we consider the plane $\mathbf{e}$ through $\mathbf{x}_0$ defined by $\mathbf{b}_1$ and $\mathbf{b}_2$, and project the vectors $\mathbf{p}(\mathbf{e})$ into $\mathbf{e}$. Doing so we get a 2D vector field $\mathbf{q}$ in $\mathbf{e}$ which has a critical point in $\mathbf{x}_0$. Figure 13a illustrates the definition of $\mathbf{q}$. To get the flow behavior of $\mathbf{p}$ in $\mathbf{x}_0$, we can classify $\mathbf{x}_0$ in $\mathbf{q}$ by an eigenvalue analysis of the Jacobian matrix $\mathbf{J_q}$ of $\mathbf{q}$ at $\mathbf{x}_0$. Computing the eigenvalues e_1, e_2 of $\mathbf{J_q}(\mathbf{x}_0)$ we get by a straightforward exercise in algebra



(a) Attracting. (b) Repelling. (c) Saddle-like.

Fig. 12. Behavior of stream lines of $\mathbf{p}$ starting on a circle $\mathbf{k}$ around $\mathbf{x}_0$.

$$e_1\,e_2 = \frac{\det(\mathbf{p}, \mathrm{grad}(u), \mathrm{grad}(v))}{\mathbf{p}^2} \quad (11)$$

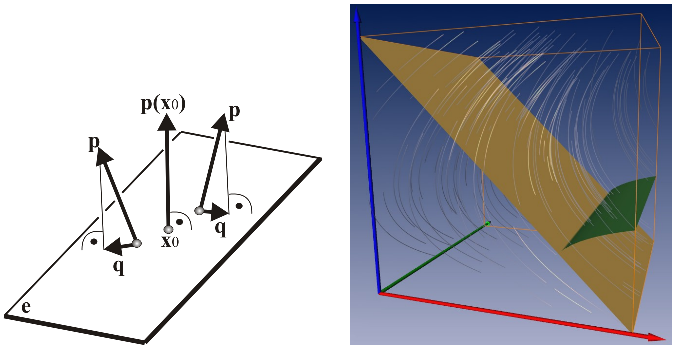
$$e_1 + e_2 = \|\mathbf{p}\| \cdot \mathrm{div}\left(\frac{\mathbf{p}}{\|\mathbf{p}\|}\right) \quad (12)$$

where $\mathrm{grad}(u) = (u_x, u_y, u_t)^T$ and $\mathrm{grad}(v) = (v_x, v_y, v_t)^T$ are the gradients of the u- and v-components respectively. [5]

To separate regions where $\mathbf{p}$ has a saddle-like behavior, we extract the isosurface $e_1\,e_2 = 0$: areas with $e_1\,e_2 < 0$ reveal a saddle-like behavior. In figure 13b as well as in the following visualizations of path line oriented topology, these surfaces are shown in a tan color. In areas with $e_1\,e_2 > 0$ we further check whether a complete attracting or repelling behavior is present. We do so by extracting and visualizing the isosurface $e_1 + e_2 = 0$: $e_1 + e_2 > 0$ denotes repelling behavior while $e_1 + e_2 < 0$ gives attracting behavior of $\mathbf{p}$. We visualize this surface in a green color. Figure 13b shows a simple example of $\mathbf{p}$ consisting of one attracting, one repelling and one saddle-like sector.

Note that this segmentation of $\mathbf{p}$ can also be achieved by considering the Gaussian and mean curvature of a surface $\mathbf{z}$ through $\mathbf{x}_0$ with all its normals in the direction of $\mathbf{p}$: a saddle-like behavior of $\mathbf{p}$ corresponds to a negative

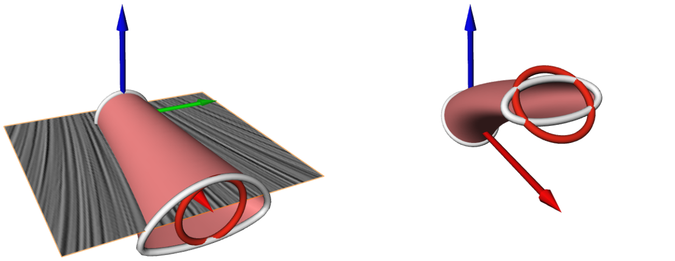
[5] Instead of eqn. (10) we can also use the local Frenet frame as a local coordinates system. However, since we apply only an eigenvalue analysis of the Jacobian matrix perpendicular to the flow direction, the results (11) and (12) are the same.



(a) Finding the projected vector field **q**.

(b) Example of **p** consisting of three sectors.

Fig. 13. Computing path line oriented topology.



(a) Stream lines show a diverging behavior.

(b) Path lines show a saddle-like behavior.

Fig. 14. Behavior of stream and path lines in the example vector field of eqn. (13) at $(0,0,0)$. As a reference, a red-colored copy of the seeding circle has been transported to the end of the stream surfaces. Comparing them with the front lines of the surfaces elucidates the behavior of the flow.

Gaussian curvature of $\mathbf{z}$, a repelling/attracting behavior corresponds to a positive/negative mean curvature.

B. Properties of path line oriented topology

In this section we collect properties of the path line oriented topology introduced above. First we show that stream lines and path lines can have a significantly different flow behavior leading to the fact that stream line and path line oriented topology reveal different characteristics of the flow. Consider the simple vector field

$$\mathbf{v}(x,y,t) = \begin{pmatrix} 1+6\,t \\ y \end{pmatrix} \tag{13}$$

in the domain $[-1,1]^3$. Analyzing the behavior of stream lines and path lines at the location $(0,0,0)$ in space-time, we obtain a diverging behavior of the stream lines. Figure 14a illustrates the seeding of a stream surface of $\mathbf{s}$ at a small circle around $(0,0,0)$ in the plane perpendicular to $\mathbf{s}(0,0,0)$. In addition, a LIC plane through $(0,0,0)$ shows the diverging behavior. Contrary to this, the path lines through $(0,0,0)$ reveal a saddle-like behavior, as shown in figure 14b. Here the seeding line is a small circle in the plane perpendicular to $\mathbf{p}(0,0,0)$.

Contrary to the stream line oriented topology, the path line oriented topology is not invariant under scalings of $\mathbf{v}$. In fact, the path lines of two vector fields $\mathbf{v}(x,y,t)$ and $c \cdot \mathbf{v}(x,y,t)$ with $c > 0$ and $c \neq 1$ differ. The factor c influences the impact of the temporal changes with respect to the spatial changes. For large c, the flow is dominated by the spatial changes. For $c \to +\infty$, the stream lines of $\mathbf{p}$ converge to the stream lines of $\mathbf{s}$.

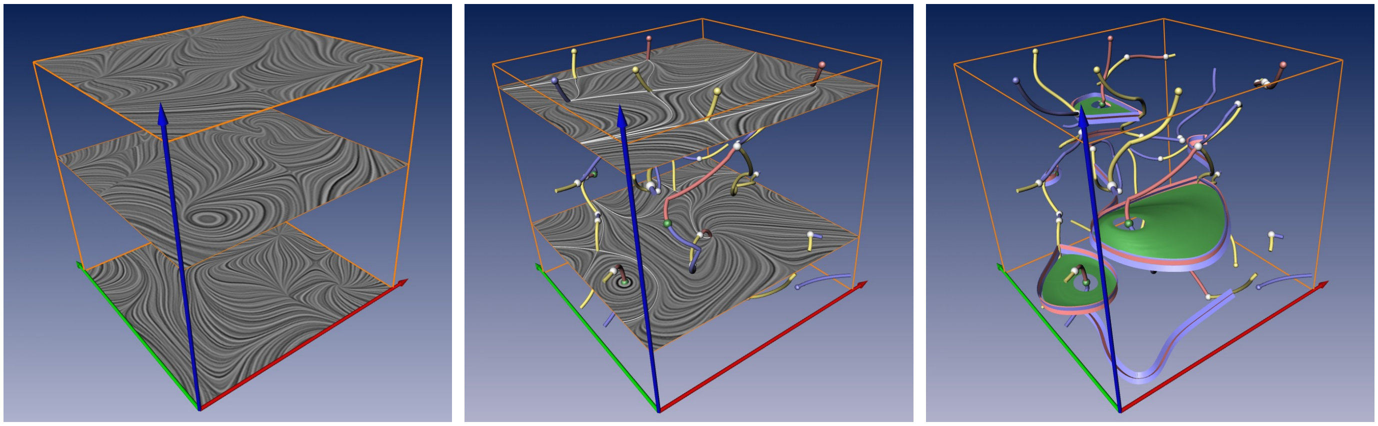
Note that the isosurfaces $e_1\, e_2 = 0$ and $e_1 + e_2 = 0$ separating different sectors in $\mathbf{p}$ are not stream surfaces of $\mathbf{p}$. This means that – contrary to stream line oriented topology – a stream line in $\mathbf{p}$ can travel through sectors of different flow behavior.

The path line oriented topology introduced above uses the concept of topology in a slightly different way than usually done. The classical understanding of topology is to observe how stream lines behave under an integration ”until infinity”, while our method only considers local properties of the path lines. However, since our approach also aims in segmenting the domain into areas of different flow behavior, we call it a topological approach as well.

V. Application and Results

We applied stream line oriented and path line oriented methods to a number of test data sets. Not surprisingly, not all topological features appear in all data sets, and different topological features turned out to be important for different data sets.

Figures 15 and 16 show a stream line oriented topological visualization of a random 2D time-dependent data set on a $5 \times 5 \times 5$ grid. Random vector fields are useful tools for a proof-of-concept of topological methods, since they contain a maximal amount of topological information. Figure 15a shows LIC images of the vector field at three different time slices which already indicates a high topological complexity. Figure 15b shows parts of the stream line oriented topological skeleton. We detected 18 critical lines of $\mathbf{s}$ (shown in red/blue/yellow, according to their outflow/inflow/saddle behavior), 32 fold bifurcations (gray spheres), and 4 Hopf bifurcations (green spheres). In this figure we also included two LIC planes to illustrate the relation between the critical lines of $\mathbf{s}$ and the critical points of $\mathbf{v}$. In addition, figure 15c shows 8 detected saddle connections, among them 4 periodic blue sky bifurcations. We visualized them as red/blue double flow ribbons describing the orientation of the intersecting separation surfaces which create them. Starting from the Hopf bifurcations, we tracked the



(a) LIC images at 3 different time slices.

(b) Tracking the locations of critical points as stream lines (red/blue/yellow); local bifurcations: Hopf bifurcations (green spheres), fold bifurcations (gray spheres).

(c) Global bifurcations: saddle connections (red/blue flow ribbons), tracked closed stream lines (green surfaces).

Fig. 15. Test data set: Stream line oriented topology of a 2D time-dependent vector field.

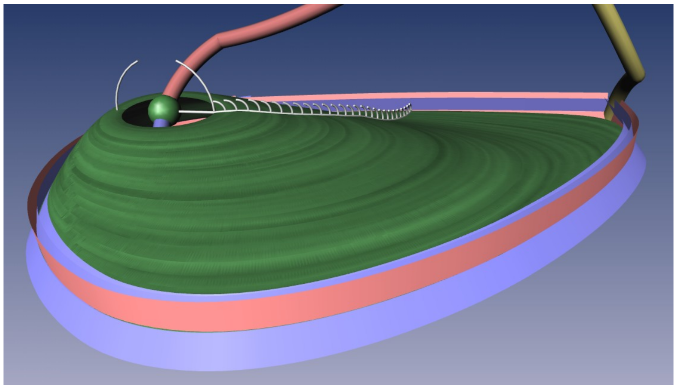


Fig. 16. Test data set: Closeup of a tracked closed stream line.

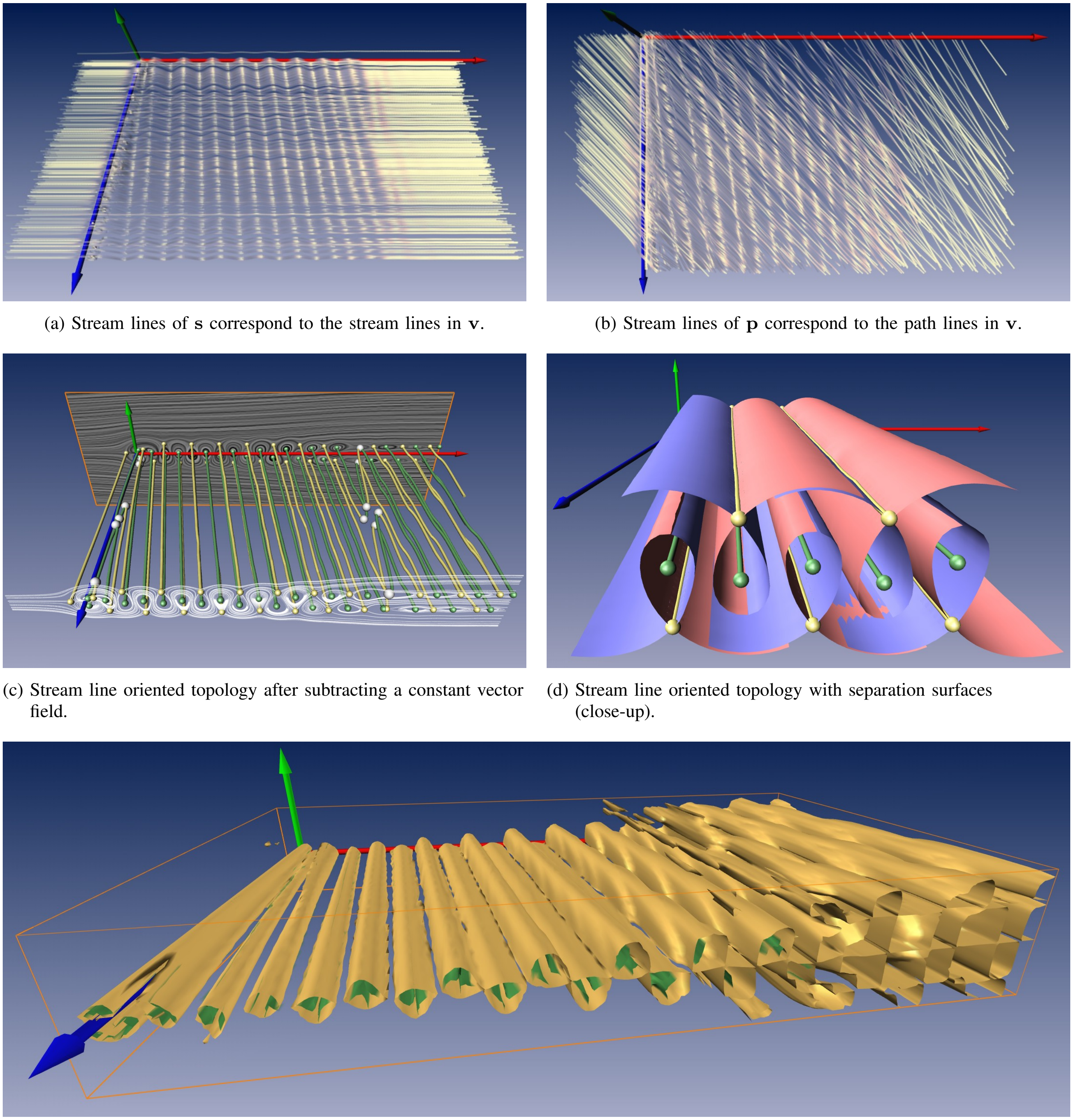
closed stream lines of $\mathbf{s}$: each closed stream line starting in a Hopf bifurcation turned out to end in a periodic blue sky bifurcation. The resulting surfaces are shown in green. Figure 16 shows a detail of figure 15c to illustrate the tracking of closed stream lines: also shown are the seeding arcs for each step of the integration.

The computing time for extracting the saddle connections in this example was 20 seconds on a Pentium 4 1.7 GHz. For this, 42 attracting and 42 repelling stream surfaces had to be integrated and checked for intersections. For tracking the closed stream lines, our algorithm took 14 seconds on the same hardware. For this, 52 steps of the described algorithm had been carried out.

Figure 17 shows the visualization of a 2D time-dependent flow behind a circular cylinder. The cylinder is in the (x,y) plane around the origin of the underlying coordinates system. This data set was kindly provided by Gerd Mutschke (FZ Rossendorf) and Bernd R. Noack (TU Berlin). Figures 17a and 17b show the stream lines of $\mathbf{s}$ and $\mathbf{p}$ as illuminated stream lines [29]. As we can see in figure 17a, this incompressible flow does not contain critical points (except for a center and a saddle directly behind the cylinder). This means that stream line oriented techniques fail for this data set. However, path line oriented techniques are still applicable. Figure 17e shows the path line oriented skeleton which reveals quasi-periodic structures behind the cylinder. These structures move slowly away from the cylinder over time. They correspond to the well-known von Kármán vortex street. To make this data set applicable to stream line oriented techniques, we subtract a constant vector field[6] which leads to the appearance of critical points. Figure 17c shows the stream line oriented skeleton of this modified data set which consists of 47 critical lines and 13 fold bifurcations. Note that the critical lines appear only in green and yellow, indicating that only moving centers and saddles are present. This corresponds to the fact that the vector field describes an incompressible flow. Figure 17c also shows that the critical points slowly move away from the cylinder over time: the critical lines of $\mathbf{s}$ are in general not parallel to the time axis. Figure 17d shows a close-up, where the separation surfaces emanating from the moving saddles are visualized. As we can see here, a number of separation surfaces tend to coincide making it impossible to extract isolated saddle connections. This is also due to the fact that the vector field is incompressible.

Figure 17 shows that for our particular data set stream line and path line oriented topology reveal rather similar

[6] This trick is well-known in the fluid dynamics community. It is motivated by the idea that the observer is moving with the flow.



(a) Stream lines of $\mathbf{s}$ correspond to the stream lines in $\mathbf{v}$.

(b) Stream lines of $\mathbf{p}$ correspond to the path lines in $\mathbf{v}$.

(c) Stream line oriented topology after subtracting a constant vector field.

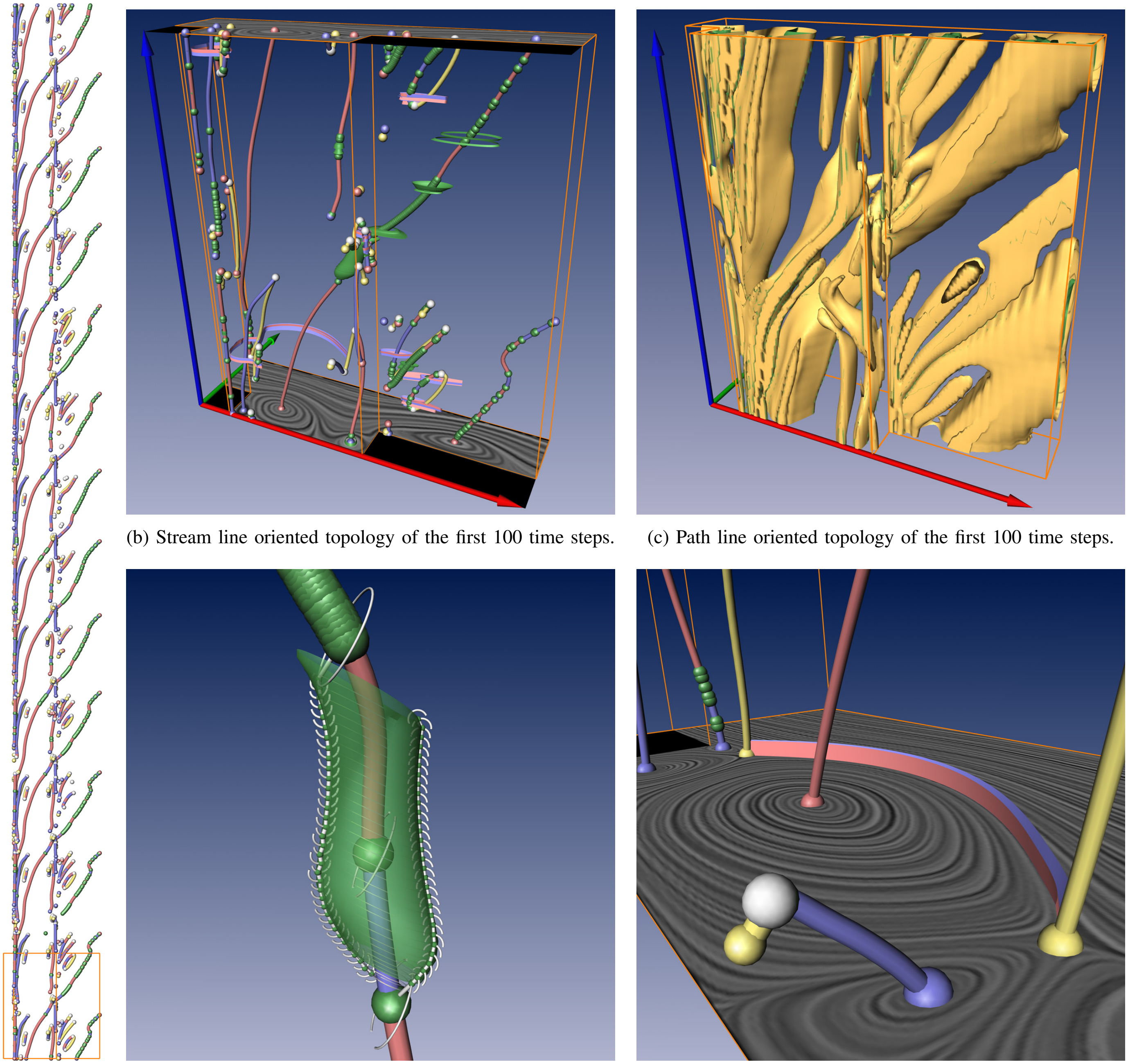
(d) Stream line oriented topology with separation surfaces (close-up).

(e) Path line oriented topology.

Fig. 17. 2D Flow behind a circular cylinder.

structures. We see the reason of this in the fact that the temporal changes in this data set are marginal in comparison to the spatial changes which dominate this flow.

Figure 18 shows the visualization of a vector field describing the flow over a 2D cavity. This data set was kindly provided by Mo Samimy and Edgar Caraballo (both Ohio State University) [30] as well as Bernd R. Noack and Ivanka Pelivan (both TU Berlin). 1000 time steps have been simulated using the *compressible* Navier-Stokes equations; it exihibits a non-zero divergence inside the cavity, while outside the cavity the flow tends to have a quasi-divergence-free behavior. The topological structures of the full data set visualized in Figure 18a



(b) Stream line oriented topology of the first 100 time steps.

(c) Path line oriented topology of the first 100 time steps.

(a) 1000 time steps.

(d) Tracked closed stream line starting and ending in a Hopf bifurcation. Tracking from both sides yields the same result as shown by the searching arcs $\hat{\mathbf{k}}$ (section III-C).

(e) Detail view with a saddle connection and a fold bifurcation.

Fig. 18. 2D time-dependent flow at a cavity. The datasets consists of 1000 time steps which have been visualized in (a). All other images show the first 100 time steps.

elucidate the quasi-periodic nature of the flow. Figures 18b-c show approximately one period – 100 time steps – of the full data set, while Figures 18d-e point out some topological details.

Figures 18b-c both reveal the overall movement of the topological structures – the most dominating ones originating in or near the boundaries of the cavity itself. The quasi-divergence-free behavior outside the cavity is affirmed by the fact that a high number of Hopf bifurcations has been found in this area. The tracked closed stream line in Figure 18d starts in a Hopf bifurcation and ends in another one – thereby enclosing a third Hopf. Our algorithm tracked it starting from both sides. The searching arcs $\hat{\mathbf{k}}$ as well as both resulting seeding lines $[\mathbf{x}_0, \ldots, \mathbf{x}_n]$ have been visualized to show the paths that have been taken by our method. Figure 18e shows a detailed view of time step 22, where a saddle connection has been detected. In the front of this figure a sink is going to join and disappear with a saddle, which just happened to enter at the domain boundary.

VI. Conclusions

In this paper we made the following contributions:

- We introduced an approach to extracting all saddle connections of a 2D time-dependent vector field.
- We introduced an approach to track closed stream lines which is robust against cyclic fold bifurcations.
- We introduced an approach to a path line oriented topology by distinguishing sectors of attracting, repelling and saddle-like behavior of the path lines.

The application to a number of test data sets shows that stream line and path line oriented topology reveal different structural properties of vector fields. While the stream line oriented topology is dominated by the presence and movements of critical points, path line oriented topology can be interpreted as focusing on mixing properties of a flow: areas with a saddle-like behavior of the path lines indicate a good mixing property of the flow over time. In particular, path line oriented topology also gives a segmentation in areas where a stream line oriented topology fails due to the absence of critical points.

While a stream line oriented topology is focused on analyzing the behavior of an integration until infinity, our path line oriented approach exclusively treats local properties of path lines. As future research we intend to explore approaches which are based on the behavior of the path lines over certain (limited) time intervals.

Acknowledgment

We thank Bernd R. Noack and Ivanka Pelivan for the fruitful discussions and the supply of simulation data sets which were kindly provided by Gerd Mutschke (cylinder) as well as Mo Samimy and Edgar Caraballo (cavity).

All visualizations in this paper have been created using Amira – a system for advanced 3D visualization and volume modeling [31] (see `http://amira.zib.de/`).

References

[1] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," *IEEE Computer*, vol. 22, no. 8, pp. 27–36, August 1989.

[2] G. Scheuermann, H. Krüger, M. Menzel, and A. Rockwood, "Visualizing non-linear vector field topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 2, pp. 109–116, 1998.

[3] W. de Leeuw and R. van Liere, "Collapsing flow topology using area metrics," in *Proc. IEEE Visualization '99*, 1999, pp. 149–354.

[4] 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.

[5] W. de Leeuw and R. van Liere, "Visualization of global flow structures using multiple levels of topology," in *Data Visualization 1999. Proc. VisSym 99*, 1999, pp. 45–52.

[6] X. Tricoche, G. Scheuermann, and H. Hagen, "A topology simplification method for 2D vector fields," in *Proc. IEEE Visualization 2000*, 2000, pp. 359–366.

[7] ——, "Continuous topology simplification of planar vector fields," in *Proc. Visualization 01*, 2001, pp. 159 – 166.

[8] R. Westermann, C. Johnson, and T. Ertl, "Topology-preserving smoothing of vector fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 3, pp. 222–229, 2001.

[9] S. Lodha, J. Renteria, and K. Roskin, "Topology preserving compression of 2D vector fields," in *Proc. IEEE Visualization 2000*, 2000, pp. 343–350.

[10] H. Theisel, "Designing 2D vector fields of arbitrary topology," *Computer Graphics Forum (Eurographics 2002)*, vol. 21, no. 3, pp. 595–604, 2002.

[11] A. Globus, C. Levit, and T. Lasinski, "A tool for visualizing the topology of three-dimensional vector fields," in *Proc. IEEE Visualization '91*, 1991, pp. 33–40.

[12] H. Löffelmann, H. Doleisch, and E. Gröller, "Visualizing dynamical systems near critical points," in *Spring Conference on Computer Graphics and its Applications*, Budmerice, Slovakia, 1998, pp. 175–184.

[13] K. Mahrous, J. Bennett, G. Scheuermann, B. Hamann, and K. Joy, "Topological segmentation in three-dimensional vector fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 2, pp. 198–205, 2004.

[14] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel, "Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields," in *Proc. IEEE Visualization 2003*, 2003, pp. 225–232.

[15] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel, "Boundary switch connectors for topological visualization of complex 3d vector fields," in *Data Visualization 2004. Proc. VisSym 04*, 2004.

[16] X. Tricoche, G. Scheuermann, and H.Hagen, "Topology-based visualization of time-dependent 2D vector fields," in *Data Visualization 2001. Proc. VisSym 01*, 2001, pp. 117–126.

[17] X. Tricoche, T. Wischgoll, G. Scheuermann, and H.Hagen, "Topology tracking for the visualization of time-dependent two-dimensional flows," *Computers & Graphics*, vol. 26, pp. 249–257, 2002.

[18] T. Wischgoll, G. Scheuermann, and H. Hagen, "Tracking closed stream lines in time-dependent planar flows," in *Proc. Vision, Modeling and Visualization 2001*, 2001, pp. 447–454.

[19] H. Theisel and H.-P. Seidel, "Feature flow fields," in *Data Visualization 2003. Proc. VisSym 03*, 2003, pp. 141–148.

[20] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel, "Stream line and path line oriented topology for 2D time-dependent vector fields," in *Proc. IEEE Visualization 2004*, 2004, pp. 321–328.

[21] J. Helman and L. Hesselink, "Visualizing vector field topology in fluid flows," *IEEE Computer Graphics and Applications*, vol. 11, pp. 36–46, May 1991.

[22] L. Abraham and K. Shaw, *Dynamics, The Geometry of Behaviour*. Addison-Wesley, 1992.

[23] P. G. Bakker, *Bifurcations in Flow Patterns (Theory and Applications of Transport in Porous Media)*. Kluwer Academic Publishers, 1991.

[24] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel, "Grid-independent detection of closed stream lines in 2D vector fields," in *Proc. Vision, Modeling and Visualization 2004*, 2004.

[25] G. Haller, "Finding finite-time invariant manifolds in two-dimensional velocity fields," *Chaos*, vol. 10, no. 1, pp. 99–108, 2000.

[26] ——, "Distinguished material surfaces and coherent structures in three-dimensional fluid flows," *Physica D*, vol. 149, pp. 248–277, 2001.

[27] R. Panton, *Incompressible Flow*. New York, etc.: John Wiley & Sons, 1984.

[28] W. de Leeuw and J. van Wijk, "A probe for local flow field visualization," in *Proc. IEEE Visualization '93*. Los Alamitos: IEEE Computer Society Press, 1993, pp. 39–45.

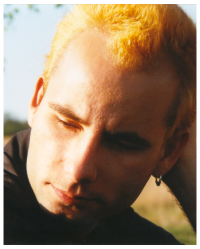
[29] M. Zöckler, D. Stalling, and H. Hege, "Interactive visualization of 3D-vector fields using illuminated stream lines," in *Proc. IEEE Visualization '96*, 1996, pp. 107–113.

[30] E. Caraballo, M. Samimy, and D. J, "Low dimensional modeling of flow for closed-loop flow control," AIAA Paper 2003-0059.

[31] D. Stalling, H.-C. Hege, and M. Westerhoff, "Amira – a highly interactive system for visual data analysis," in *Visualization Handbook*, C. R. Johnson and C. D. Hansen, Eds. Academic Press, 2004.

Holger Theisel received his M.S. (1994), Ph.D. (1996) and habilitation (2001) degrees from the University of Rostock (Germany) where he studied Computer Science (1989 – 1994) and worked as a research and teaching assistant (1995 – 2001). He spent 12 months (1994 – 1995) as a visiting scholar at Arizona State University (USA), and 6 months as a guest lecturer at ICIMAF Havana (Cuba). Since 2002 he has been a member of the Computer Graphics group at MPI Informatik Saarbrücken (Germany).

His research interests focus on flow and volume visualization as well as on CAGD, geometry processing and information visualization.



Tino Weinkauf studied computer science with the focus on computer graphics at the University of Rostock, Germany, where he received his M.S. degree in 2000. Since 2001 he has been performing research as a Ph.D. student at the Scientific Visualization department of Zuse Institute Berlin (ZIB). He is associated with the Collaborate Research Center (Sfb 557) "Control of Complex Turbulent Flows", where he works on feature based analysis and comparison techniques for flow fields. His current research interests focus on flow and tensor analysis, information visualization and visualization design.

Hans-Christian Hege is head of the Scientific Visualization department at Zuse Institute Berlin (ZIB). After studying physics and mathematics he performed research in computational physics and quantum field theory at Freie Universität Berlin (1984-89). Then he joined ZIB, initially as scientific consultant for high performance computing then as head of the Scientific Visualization department, which he started in 1991. His group performs research in data visualization and develops visualization software, like e.g. the Amira software. He is also co-founder of mental images (1986) and of Indeed - Visual Concepts GmbH (1999) which he built up as CEO (-2003) and which he now consults as scientific adviser. He regularly lectures at Universität Pompeu Fabra, Barcelona (Spain) and the German Film School (University for Digital Media Production) in Elstal, where he became honorary professor of Scientific Visualization in 2003. His current research interests are in visual data analysis and computer graphics as well as applications in engineering, natural and life sciences.

Hans-Peter Seidel studied mathematics, physics and computer science at the University of Tübingen, Germany. He received his Ph.D. in mathematics in 1987, and his habilitation for computer science in 1989, both from University of Tübingen. From 1989, he was an assistant professor at the University of Waterloo, Canada. In 1992, he was appointed to the chair of Computer Graphics of the University of Erlangen-Nürnberg, Germany.

Since 1999 he has been director of the Computer Graphics Group at the Max-Planck-Institute for Computer Science and honorary professor at Saarland University in Saarbrücken, Germany. In his research, Hans-Peter Seidel investigates algorithms for 3D Image Analysis and Synthesis. This involves the complete processing chain from data acquisition over geometric modeling to image synthesis. He was awarded the Leibniz prize 2003 – the highest scientific award in the German system.