Streamline Predicates as Flow Topology Generalization

Tobias Salzbrunn and Gerik Scheuermann

University of Leipzig
{salzbrunn,scheuermann}@informatik.uni-leipzig.de

Summary. Streamline predicates are simply boolean functions on the set of all streamlines in a flow field. A characteristic set of a streamline predicate is the set of all streamlines fulfilling the predicate. If streamline predicates are defined based on asymptotic behavior, the characteristic sets become α- or ω-basins. Using boolean algebra on the streamline predicates, we obtain the usual flow topology. We show that these considerations allow us to generalize flow topology to flow structure definitions. These flow structure definitions can be flexibly adapted to typical analysis tasks arising in flow studies and taylored to the users' needs

1 Introduction

Flow topology has been developed into a tool that gives information about the course of streamlines in steady two and three-dimensional velocity vector fields. Basically, it clusters streamlines with similar behavior. The clustering is based on a precise definition, namely the basins of dynamical system theory. Therefore, each cluster can be interpreted clearly by the user. This is one of the advantages of topology compared to other clustering methods, like typical statistical clustering [6], [16], anisotropic diffusion [13] or an algebraic multigrid approach [5].

But it must be said that there are also limitations. One drawback is missing Galilean invariance. Topology changes between a fixed observer and an observer moving with constant velocity (because the streamlines change their course). We think that this problem can be solved in many cases by simply taking the given observer of the data. This is useful in typical flows around a single airplane, train, or other obstacle. It is also an obvious choice for flows inside, e.g., a building, cabin or turbine. In more complex situations, we suggest the use of the localized flow approach of Wiebel et al. [22] that allows to remove any flows crossing the outer boundary and is Galilean invariant without creating flow through solid boundaries like the popular method of removing the average flow.

Another, in our eyes more important, limitation of topology is that it may miss relevant aspects of streamline behavior. The most important example are vortices (areas of high vorticity) in three-dimensional flows. Quite often, topology groups streamlines obviously entering the vortex and streamlines not passing the vortex region into the same group because they belong to the same basin in the sense of dynamical system theory. We give a realistic example in the results section. Similar problems can arise with streamlines crossing shocks or entering shear flow areas. Since engineers and scientists often like to distinguish streamlines entering and not entering a vortex, we suggest a solution in this paper that refines topology in these cases.

There is a further limitation of steady flow topology that can hinder understanding: topology does not depend on absolute velocity. Topology concentrates on the set of points visited by a streamline but the visit time does not play a role. But sometimes, engineers are interested only in fast dynamics or the time a particle resides near a surface. We will show that these concepts can be easily expressed by streamline predicates and could therefore be used to enrich topology.

2 Related Work

Of course, this paper builds on quite a large number of publications in flow topology, especially [7, 9, 14, 15, 23, 18, 10, 17, 21, 3]. Besides, we have been also influenced by feature-based visualization [12], especially the early work of van Walsum et al. [19], the work on vortex detection by Peikert et al. [11, 1] and the feature definition language of Doleisch and Hauser [2]. In the previous section, we have already mentioned relevant articles on cluster-based flow visualization.

3 Streamline Predicates

We concentrate our consideration on steady three-dimensional flows. Of course, the planar case is quite similar. Let $D \subset R^3$ be the ***domain***. A ***vector field*** on D is a Lipschitz continuous map

$$\begin{aligned} v : D &\to R^3, \\ x &\mapsto v(x). \end{aligned}$$

A ***streamline of v passing through the point*** $a \in \mathbf{D}$ is a continuous map

$$s_a : J_a \to D$$

where $0 \in J_a \subset R$ is an interval of maximal extend and s_a fulfills the conditions

$$s_a(0) = a,$$
$$\dot{s_a}(\tau) = v(s_a(\tau)) \;\; \forall \tau \in J_a.$$

Since we are interested in the set of streamlines, we identify two streamlines $s_a \simeq s_b$ if there is a $\tau_0 \in R$ such that

$$s_a(\tau) = s_b(\tau + \tau_0)$$

and note that $s_a \simeq s_b$ if there are $\tau \in J_a,\ \tau' \in J_b$ with $s_a(\tau) = s_b(\tau')$ due to the existence and uniqueness theorem for streamlines. We define the set of all streamlines as the set $\mathcal{S}$ of all equivalence classes.

Let $s_\lambda : J_\lambda \to D$ be a representive of $S_\lambda \in \mathcal{S}$. Every other representive could then be written as $s'_\lambda : J_\lambda + \tau_0 \to D,\ s'_\lambda(\tau + \tau_0) = s_\lambda(\tau)$. Since the set of points $s_\lambda(J_\lambda)$ (the course of the streamline) is the same for equivalent streamlines, we can define it as $S_\lambda(J_\lambda) := s_\lambda(J_\lambda)$. Then, we have a partition of $D = \bigcup_{S_\lambda \in \mathcal{S}} S_\lambda(J_\lambda)$, since the equivalence classes are mutually disjoint.

A ***streamline predicate*** is defined as a map

$$SP: \ \mathcal{S} \to \{\ TRUE, FALSE\ \},$$
$$S \mapsto SP(S).$$

i.e. a boolean map on the streamlines that does not depend on the absolute time at only one position. It may nevertheless depend on relative time between different positions along the streamline.

The ***characteristic set*** of a streamline predicate is defined as

$$C_{SP} := \bigcup_{S_\lambda \in S,\ SP(S)=TRUE} S_\lambda(J_\lambda) \subset D.$$

4 Flow Structure

Our goal in this paper is a definition of flow structure that meets the needs of users in all cases and extends flow topology. A flow structure is considered a partition of the flow into disjunct clusters. We suggest a partition based on streamlines. This agrees with the approach taken by flow topology. Since we want to have a general grouping mechanism, we start with a finite set $\mathcal{G}$ of streamline predicates

$$\mathcal{G} = \{\ SP_\lambda \mid \lambda \in \Gamma\ \}$$

which is chosen such that their characteristic sets are disjoint, i.e.

$$C_{SP_\lambda} \ \cap\ C_{SP_\mu} = \emptyset \ \ \forall \lambda, \mu \in \Gamma.$$

As ***flow structure***, we define the partition of

$$\mathcal{S} = \bigcup C_{SP_\lambda}$$

where we assume that every streamline fulfills exactly one streamline predicate. If $\mathcal{G}$ creates only a partial partition of $\mathcal{S}$, we add the predicate

$$\begin{aligned} SP_0 : \ & \mathcal{S} \to \{ \ TRUE, FALSE \ \}, \\ & S \mapsto \bigwedge_{\lambda \in \Gamma} \{ \ SP_\lambda(S) = FALSE \ \}. \end{aligned}$$

In the next section we show that the usual flow topology is a special flow structure.

5 Flow Topology as Flow Structure

Following Scheuermann et al. [15], we define topology using α- and ω-limit sets. For a streamline s, we define its ***α-limit set*** $A(s)$ as

$$A(s) := \{ \ p \in R^3 \mid \exists (t_n)_{n=0}^{\infty} \subset R, \ t_n \to -\infty, \ \lim_{n \to \infty} s(t_n) = p \ \}$$

and its ***ω-limit set*** as

$$\Omega(s) := \{ \ p \in R^3 \mid \exists (t_n)_{n=0}^{\infty} \subset R, \ t_n \to \infty, \ \lim_{n \to \infty} s(t_n) = p \ \}.$$

If a streamline enters or leaves the domain D at the boundary ∂D, we define the boundary ∂D as α- resp. ω-limit set.

The union of all streamlines with α-limit set A is called the ***α-basin of A***

$$B_\alpha(A) = \{ \ a \in D \mid A(s_a) = A \ \}.$$

Similarly, the union of all streamlines with ω-limit set Ω is called the ***ω-basin of Ω***

$$B_\omega(\Omega) = \{ \ a \in D \mid \Omega(s_a) = \Omega \ \}.$$

If $A_i, \ i \in I$, and $\Omega_j, \ j \in J$, denote all α- and ω-limit sets in D and $Z_k(M)$ denotes the connected components of $M \subset D$, the ***flow topology of v*** can be described as the partition

$$D = \bigcup_{i,j,k} Z_k(B_\alpha(A_i) \cap B_\omega(\Omega_j)).$$

In our framework, we use the following predicates

$$\begin{aligned} SP_{A_i} : \ & \mathcal{S} \to \{ \ TRUE, FALSE \ \}, \ i \in I, \\ & S \mapsto A(S) = A_i. \end{aligned}$$

$$SP_{\Omega_j}: \mathcal{S} \to \{ \mathit{TRUE}, \mathit{FALSE} \}, \ j \in J, \\ S \mapsto \Omega(S) = \Omega_j$$

since α- and ω-basins are the same for equivalent streamlines. We get the basins of the topology as characteristic sets, i.e.

$$C_{SP_{A_i}} = B_\alpha(A_i) \quad C_{SP_{\Omega_i}} = B_\omega(\Omega_i).$$

Therefore, we can use the set of predicates

$$\mathcal{G}_{TOP} = \{ SP_{A_i} \ \mathit{AND} \ SP_{\Omega_j} \mid i \in I, \ j \in J\},$$

as definition of a flow structure that coincides with flow topology.

6 Refinement of Flow Topology

Looking at section 5, we can ask what is gained by using streamline predicates and general flow structures compared to flow topology. The answer is a wide flexibility because there is no reason to choose exactly the predicates used in the previous section.

We want to show this flexibility using an important example in practice. A user studies steady flow around an obstacle (car, airplane, train, sphere, ellipsoid, house, ...) and is interested in vortices. For the flow topology, critical points, closed streamlines, and boundary switch points are determined. As next step, separating surfaces and isolated streamlines starting at saddle points are computed. Including an analysis of the boundary of the obstacle, it is likely that even for vortices close to a typical model like Vatistas [20], the vortex will show up only as a single streamline. Streamlines obviously rotating around this line and streamlines not rotating around the line will be in the same topological component. At this point, streamline predicates can show their strength. In a first step, the user can apply any vortex detection method, e.g. the λ_2-method of Jeong and Hussein [8], and define the extend of vortices. In a second step, he defines a streamline predicate for each vortex that decides if the streamline crosses the vortex region. The third step creates a flow structure using all and-combinations of the SP_{A_i} *and* SP_{Ω_j} of predicates from topology with the vortex predicates and their opposite predicates. In this way, streamlines entering the vortex are distinguished from streamlines missing it.

Of course, whenever the user defines interests in the behavior of streamlines with streamline predicates, a similar solution is possible. Therefore, flow structure based on streamline predicates allows a refinement of flow topology tailored to the users needs.

7 Results

In the remainder of this paper we want to present three examples of streamline predicates addressing questions which could not be answered with flow topology methods. The dataset we use corresponds to a single time step of an unsteady simulation of the German train ICE. The train travels at a velocity of about 250 km/h with a wind blowing from the side at an angle of 30 degrees. The wind causes vortices to form on the lee side of the train, creating a drop in pressure that has adverse effects on the trains track holding. For our computations we choose a region of interest around the front wagon. To represent the set of all streamlines $\mathcal{S}$ we choose a finite subset $\tilde{\mathcal{S}}$. We use a Cartesian grid in the area $[-15000, 45000] \times [-15000, 25000] \times [350, 5500]$ with 200 units as spacing in all directions as starting positions for the streamlines in $\tilde{\mathcal{S}}$. This is a set of more than 1.56 million streamlines that fills the space around the train in a dense manner.

The first predicate is exemplary for streamline predicates using time information of steady vector fields (i.e. absolute velocity). We are interested in parts of the flow which have a direct influence on the surface (and immediate neighborhood) of the train. Especially particles residing a "long" time near the surface are of interest. Of course one could use a fixed minimum residence time given by some physical considerations for a given application area. However we take another approach and calculate the residence time for a representative set of streamlines to get an idea of a meaningful value. From the resulting distribution we take the value of the 99-% quantile as minimum residence time t_{min}. For the required minimum distance calculation we compute a distance field on the positions of the dataset grid thus reducing minimum distance calculations to a simple interpolation in the distance field at a questioned position. Fig. 1 shows the isosurface of the distance field for an isovalue of 20 [cm] (which we use as maximal neighborhood distance for our computations). We define the following general streamline predicate (instantiated with the previous values):

$$A \ - \tilde{S} \ stays \ a \ minimum \ time \ t_{min} \ in \ the \ neighborhood \ of \ an \ object$$

The resulting flow structure $\mathcal{G}_{Surface} = \{ A, \bar{A} \}$ is of course very simple, but will get more complex if more than one object is taken into account. Fig. 1 shows the boundary of the resulting characteristic set A. There is one part of the flow hitting the train on the luv side and flowing around the train and a second part hitting the head of the train and being pushed towards the trains surface (lee side).

In the second example we want to study the deviation of the flow from the principal input flow direction thus getting the most turbulent parts of the flow. To compute the deviation we integrate the difference between the tangent vector direction and the main inflow direction along the streamlines. Again we sample a representative set of streamlines, compute the deviation and take the 99-% quantile as minimum deviation d_{min}. We define the streamline predicate:

$$D \ - Deviation\ of \tilde{S}\ from\ principal\ direction\ is\ greater\ than\ d_{min}$$

Fig. 2 shows the boundary of the resulting characterist set D. One can see the flow that deviates very strong from the principal inflow direction.

In the third example we examine the interplay between vortices in the flow and the flow regions outside the vortex regions. Applicability of topological methods is limited concerning this important application domain due to the lack of singularities of the velocity field. We examine if a streamline enters a certain vortex region in order to test if a streamline is influenced by a vortex. Of course more precise and sophisticated methods are possible, but we decide to hold computational effort down. To compute the vortex regions we use the λ_2-criteria of Jeong and Hussain [8]. The λ_2-criteria does not clearly separate vortex regions of different vortices, especially if they are close together. To address this issue we compute as additional information about the vortices the vortex core lines with the gravity-line-method explained in [4]. The resulting vortex core lines are depicted in Fig. 3. Based upon the vortex core lines we use a flood-fill algorithm to label each cell according to which vortex region (if any) it belongs to. We start with the cells that inherit a segment of a vortex core line computed in the previous step. Each cell is examined for its λ_2-value. To get a cell based λ_2-value we assign to every cell the mean of the λ_2-values of its vertices. If the cell has a negative λ_2-value it gets the label of the respective vortex core. The labeled cells are put into a priority queue with the most negative λ_2-value on top. After the initial feeding of the priority queue the neighboring cells of the top element of the queue are examined for their λ_2-values. If a neighboring cell with negative λ_2-value exists it gets the label of the top element. The top element is removed afterwards and the labeled neighboring cells are inserted according to their λ_2-values. This strategy insures that regions with strong vortices grow faster. The flood-fill algorithm is finished if the queue is empty. We now have vortex cores with corresponding regions as a set of cells with the appropriate label. The vortex regions of the train-dataset are depicted in Fig. 3. To compute the following streamline predicates one has to check if a streamline enters the cells of a vortex core.

We evaluate the three streamline predicates

$$R \ - \tilde{S}\ enters\ the\ red\ vortex\ region$$
$$G \ - \tilde{S}\ enters\ the\ green\ vortex\ region$$
$$B \ - \tilde{S}\ enters\ the\ blue\ vortex\ region$$

For the flow structure, we need a set of streamline predicates with disjunct characteristic sets filling up D. Unfortunately, the lack of singularities prohibited to compute the $SP_{A_i}\ and\ SP_{\Omega_j}$ predicates from topology. Additionally attempts to start from the surface topology were not successful. We did not examine boundary switch connectors as proposed in [21], but we assume that they will not separate the vortices in a way one would expect it according to the λ_2-criteria. Therefore, we choose the set

$$\mathcal{G}_{Vortex} = \{ \bar{R} \wedge \bar{G} \wedge \bar{B},\ \bar{R} \wedge \bar{G} \wedge B,\ \bar{R} \wedge G \wedge \bar{B},\\ \bar{R} \wedge G \wedge B,\ R \wedge \bar{G} \wedge \bar{B},\ R \wedge \bar{G} \wedge B,\\ R \wedge G \wedge \bar{B}, R \wedge G \wedge B \},$$

In this way, we separate streamlines by the vortex regions they enter. $\bar{R} \wedge G \wedge B$, for example describes the streamlines entering the green and the blue vortex region, but not entering the red vortex region. Fig. 4 and 5 show the boundaries of all characteristic sets of $\mathcal{G}_{Vortex}$ (except $\bar{R} \wedge \bar{G} \wedge \bar{B}$).

8 Conclusion

We introduced streamline predicates as a new tool to study flow datasets. We showed that a flow structure based upon appropriate streamline predicates comprises and refines flow topology. Applied to one realistic CFD-dataset, streamline predicates proofed able to answer questions where conventional topological methods could not be applied. Computing the streamlines and the characteristic sets requires high computational effort for brute force implementations. Further research should deal with increasing the efficiency of the computations.

Acknowledgment

The authors wish to thank Markus Rütten, from German Aerospace Center(DLR) in Göttingen for providing the datasets. We also wish to thank Xavier Tricoche and Christoph Garth for their valuable advice. Last but not least thanks go to all members of the FAnToM development team for their programming efforts. This work was partly supported by DFG grants HA 1491/15-5 and SCHE 663/3-7.

References

1. D. Bauer and Peikert R. Vortex Tracking in Scale Space. In D. Bauer, P. Brunet, and I. Navazo, editors, *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization 2002 (EuroVis 2002)*, pages 233–240, 2002.
2. H. Doleisch, M. Gasser, and H. Hauser. Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data. In *Proceedings of the 5th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 239 – 248, 2003.
3. C. Garth, X. Tricoche, and G. Scheuermann. Tracking of vectorfield singularities in unstructured 3d-time dependent datasets. In *IEEE Visualization 2004*, pages 329 – 336, Austin, Texas, 2004.
4. C. Garth, Tricoche X., T. Salzbrunn, Bobach T., and G. Scheuermann. Surface Techniques for Vortex Visualization. In *VisSym*, pages 155–164, 346, 2004.

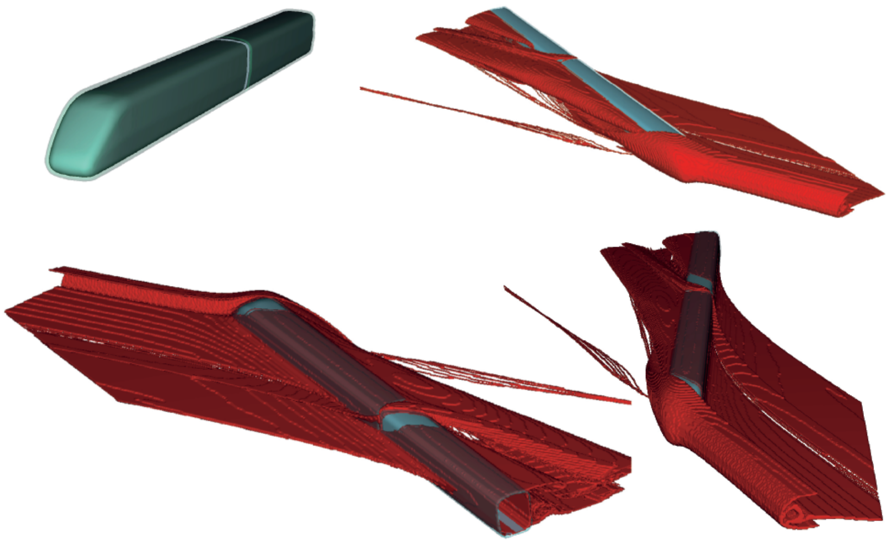

Fig. 1. Boundary of the characteristic set A of the flow structure $\mathcal{G}_{Surface}$. (Upper left picture shows the hull around the train.)

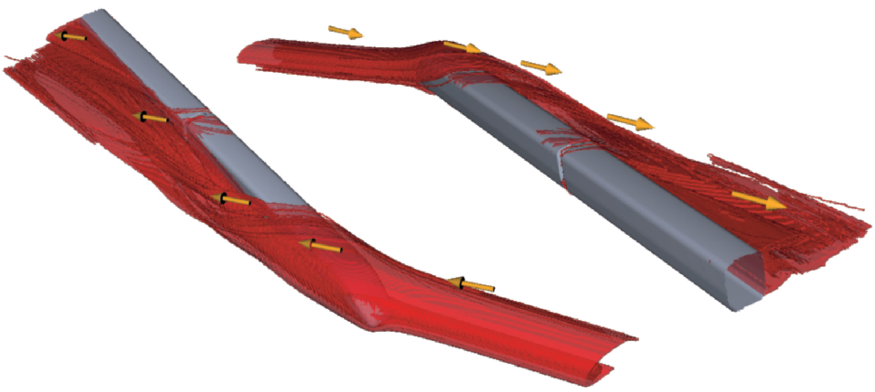

Fig. 2. Boundary of the characteristic set D of the deviation predicate. (The arrows show the principal direction of the entry flow.)

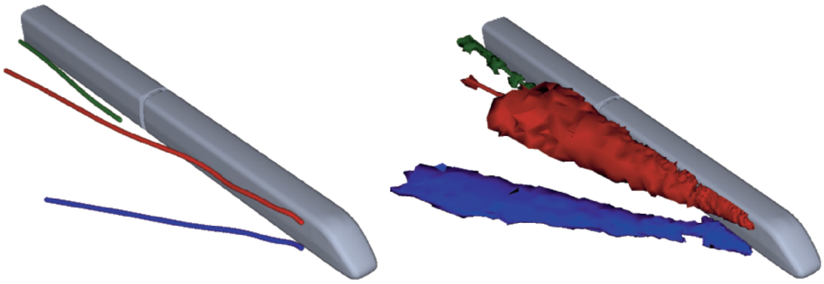

Fig. 3. The left picture shows the vortex core lines of the train-dataset. In the right picture the corresponding vortex core regions are depicted. (Note that the vortex core regions do not cover the vortex core lines completely. We assume that the vortices level off at the end.) (Colorplate on p. 206.)

5. M. Griebel, T. Preusser, M. Rumpf, M.A. Schweitzer, and A. Telea. Flow Field Clustering via Algebraic Multigrid. In *IEEE Visualization 2004*, pages 35 – 42, Austin, Texas, 2004.
6. B Heckel, G.H. Weber, B Hamann, and K.I. Joy. Construction of Vector Field Hierarchies. In *IEEE Visualization 1999*, pages 19–25, San Francisco, CA, 1999.
7. J. L. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991.
8. J. Jeong and F. Hussain. On the Identification of a Vortex. *Journal of Fluid Mechanics*, 285:69 – 94, 1995.
9. H. Löffelmann, T Kucera, and Gröller E. Visualizing Poincare Maps Together with the Underlying Flow. In H.C. Hege and K. Polthier, editors, *Proceedings of the International Workshop on Visualization and Mathematics 1997 (VisMath'97)*, pages 315–328, 1998.
10. K. Mahrous, J. Bennett, G. Scheuermann, B. Hamann, and K. I. Joy. Topological Segmentation of Three-Dimensional Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):198 – 205, 2004.
11. R. Peikert and M. Roth. The Parallel Vectors Operator - a Vector Field Visualization Primitive. In *IEEE Visualization 1999*, pages 263 – 270, San Francisco, CA, 1999.
12. F.H. Post, B. Vrolijk, H. Hauser, R.S. Laramee, and H. Doleisch. The State of the Art in Flow Visualization: Feature Extraction and Tracking. In *Computer Graphics Forum 22*, volume 4, pages 775–792, 2003.
13. T. Preusser and M. Rumpf. Anisotropic nonlinear diffusion in flow visualization. In *IEEE Visualization 1999*, pages 325–332, San Francisco, CA, 1999.
14. G. Scheuermann, Krüger H., M. Menzel, and Rockwood A. Visualizing Nonlinear Vector Field Topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109 – 116, 1998.
15. G. Scheuermann, I.J. Kenneth, and W. Kollmann. Visualizing Local Vector Field Topology. *Journal of Electronic Imaging*, 9:356–367, 2000.
16. A. Telea and J.J. van Wijk. Simplified representation of vector fields. In *IEEE Visualization 1999*, pages 35–42, San Francisco, CA, 1999.
17. 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 *IEEE Visualization 2003*, pages 225 – 232, 2003.

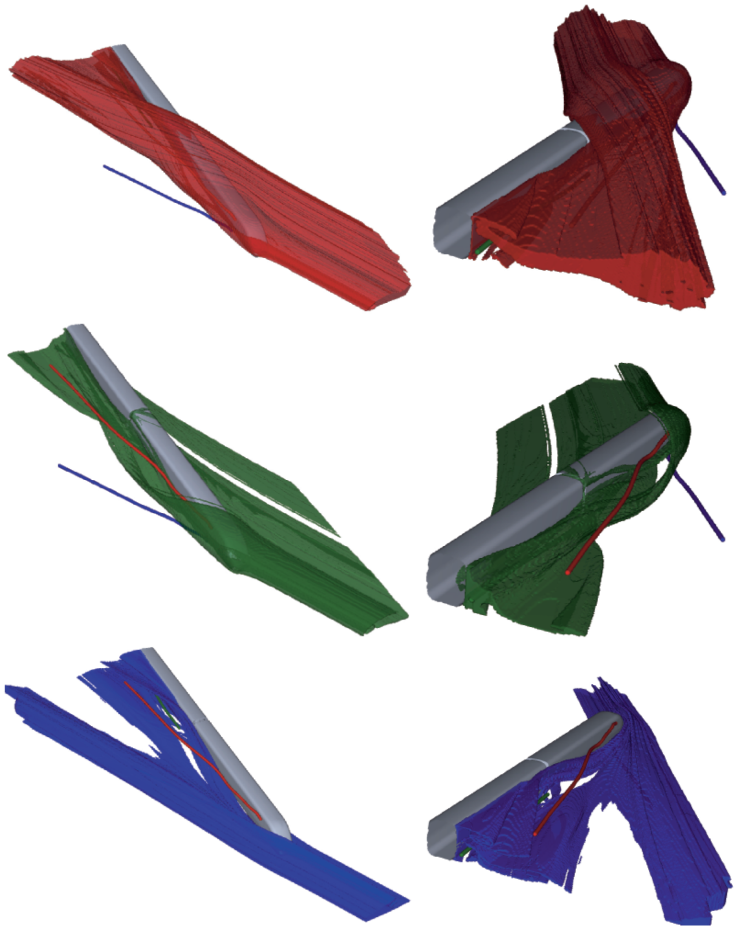

Fig. 4. Boundaries of characteristic sets of $\mathcal{G}_{Vortex}$ characterizing these parts of the flow that stream only in one vortex region (top down): $R \wedge \bar{G} \wedge \bar{B}$, $\bar{R} \wedge G \wedge \bar{B}$, $\bar{R} \wedge \bar{G} \wedge B$ (colorplate on p. 206).

18. X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topological Tracking for the Visualization of Timedependent Two-Dimensional Flows. *Computers & Graphics*, 26(2):249 – 257, 2002.
19. T. van Walsum, F. H. Post, D. Silver, and F. J. Post. Feature Extraction and Iconic Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):111 – 119, 1996.
20. G.H. Vatistas. New Model for Intense Self-Similar Vortices. *Experiments in Fluids*, 14(4):462–469, 1998.
21. T. Weinkauf, H. Theisel, H.C. Hege, and Seidel H.P. Boundary Switch Connectors for Topological Visualization of Complex 3d Vector Fields. In *Proceedings of the 6th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2004)*, pages 183 – 192, 2004.

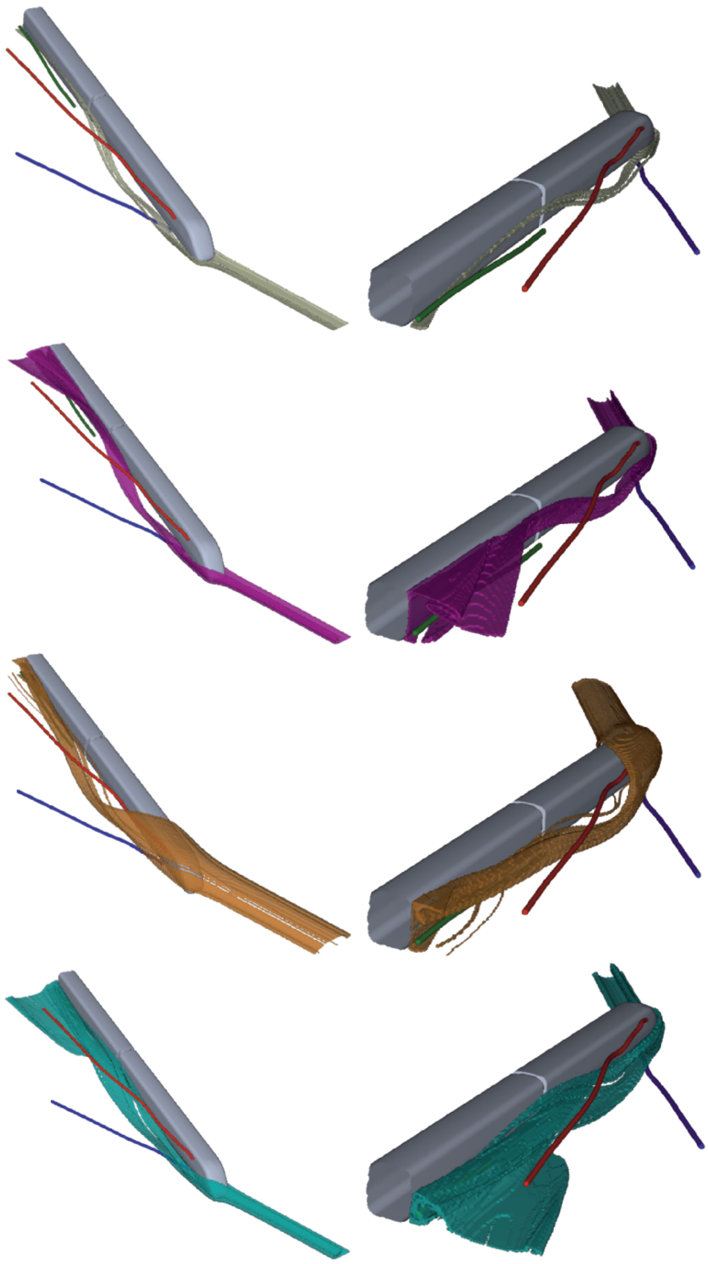

Fig. 5. Boundaries of characteristic sets of $\mathcal{G}_{Vortex}$ characterizing these parts of the flow that stream in more than one vortex region (top down): $R \wedge G \wedge B$, $R \wedge \bar{G} \wedge B$, $R \wedge G \wedge \bar{B}$, $\bar{R} \wedge G \wedge B$ ($\bar{R} \wedge \bar{G} \wedge \bar{B}$ (i.e. flowing in no vortex region) is not shown.) The second column shows the same characteristic sets from another view (colorplate on p. 207).

22. A Wiebel, C Garth, and G Scheuermann. Localized Flow Analysis of 2d and 3d Vector Fields. In Ken Brodlie, David Duke, and Ken Joy, editors, *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization 2005 (EuroVis 2005)*, pages 143–150, 2005.
23. T. Wischgoll and G. Scheuermann. Detection and Visualization of Closed Streamlines in Planar Flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165 – 172, 2001.