

Visualizing Linear Neighborhoods in Non-Linear Vector Fields

Stefan Koch*
University of Leipzig

Alexander Wiebel†
Zuse Institute Berlin (ZIB)

Jens Kasten‡
University of Leipzig

Mario Hlawitschka§
University of Leipzig

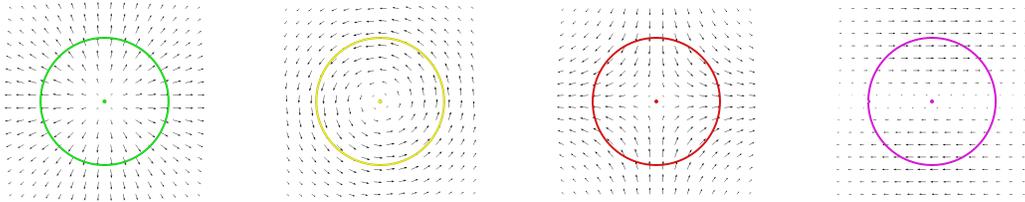


Figure 1: Linear neighborhoods computed in originally linear fields (source, center, saddle and shear) that have been multiplied with a 2D Gaussian kernel. The result is non-linear. The shape of the of the linear neighborhoods reflects the circular kernel causing the non-linearity.

ABSTRACT

Linear approximation plays an important role in many areas employing numerical algorithms. Particularly in the field of vector field visualization, it is the basis of widely used techniques. In this paper, we introduce two methods to extract areas in two- and three-dimensional vector fields that are connected to linear flow behavior. We propose a region-growing algorithm that extracts the *linear neighborhood* for a certain position. The region is characterized by linear flow behavior up to a user-defined approximation threshold. While this first method computes the size of a region given the mentioned threshold, our second method computes the quality of a linear approximation given a user-defined n -ring neighborhood. The scalar field resulting from the second method is, therefore, called *affine linear approximation error*. Isosurfaces of this field show regions of close-to-linear and non-linear flow behavior. We demonstrate the expressiveness and discuss the properties of the extracted regions using analytical examples and several datasets from the domain of computational fluid dynamics (CFD).

Index Terms: Computing Methodologies [I.3.8]: Computer Graphics—Applications, Computer Applications [J.2]: Physical Sciences and Engineering—Physics

1 INTRODUCTION

Linear vector fields are well understood since they depict the most simple class of non-trivial fields. They can be described by a single tensor, i.e., the Jacobian matrix. Besides the relevance of linear vector fields for understanding flows, linear approximations play an important role in many areas employing numerical methods. Particularly in the field of vector field visualization, they are the basis of widely used techniques. These naturally include all methods using linear interpolation, which are ubiquitous.

The importance of linear approximations is even more evident when looking at the methods that are based on the Jacobian matrix, i.e., the vector gradient. Many methods for feature detection are based on this local description of the field – among them vortex core definitions [8] and vortex core line detection methods [29, 17]. Other methods use it only in parts of their overall approach – these

include vector field topology [7, 31], Lagrangian flow visualization [5, 3, 24], and glyph-based methods [12, 1, 32]. Finally, some methods can be improved by taking such local linear descriptions into account as, e.g., the computation of stream surfaces [26] and flow separation line extraction [9].

All these methods in this incomplete list use the local linear approximation for distinguished stages in their processing pipeline. Unfortunately, the local linear approximation is mostly restricted to a cell, since it is often not clear how to extend the approximation region without violating the linearity assumption. The goal of this paper is to address this problem by providing and studying appropriate measures. So far, there are only two methods that employ the linearity of the flow in a certain *region* [26, 32]. These methods compute a radius of a sphere for which the flow can be assumed to behave sufficiently linear.

In this paper, we enhance the basic idea of the two latter methods by an identification of regions that are mainly linear describeable and provide deeper insights into the local and overall flow behavior. Therefore, we present a technique that explicitly computes a region of approximately linear behavior for every position in the field. To do so, we enhance the underlying linear approximation used by the above-mentioned methods by an affine transformation. The boundary of the *linear neighborhood* of a certain point is efficiently computed by an algorithm similar to the concept of marching cubes used for isosurface extraction. In addition to the extraction of linear neighborhoods, we introduce an error measure for the quality of linear approximation at a certain point. Analyzing an n -ring neighborhood for each point in the field, the error of a linear approximation is computed. Using isosurfacing on the resulting field, we can identify regions of close-to-linear and non-linear flow behavior. In summary, the main contributions of the paper are

- a method that (a) computes regions of approximately linear flow behavior *explicitly* (in contrast to only their extent [26, 32]) and that (b) is applicable to any position in the domain (in contrast to only to singularities [26, 32]),
- a method that quantifies the linearity of n -ring neighborhoods of every position in a dataset (see Sec. 3.2), and
- an interpretation of how these regions are related to previously known region-based characteristics of vector fields.

We use analytic examples as well as computational fluid dynamics (CFD) data from several application areas to illustrate the characteristics and interpretations of the extracted regions (see Sec. 5).

2 RELATED WORK

As the applications of the presented techniques mainly come from the field of fluid dynamics, much work on flow visualization can be

*e-mail: skoch@informatik.uni-leipzig.de

†e-mail: wiebel@zib.de

‡e-mail: kasten@informatik.uni-leipzig.de

§e-mail: hlawitschka@informatik.uni-leipzig.de

considered to be related. A good overview of the literature in this area can be found in specific state-of-the-art reports that discuss feature-based [19, 15, 11] and partition-based [25] techniques. Our methods consider connected regions of vector fields having overall similar characteristics. To be able to search for such similarities, one has to define a similarity measure or, equivalently, a measure for dissimilarity (distance, error). Such measures are well-known from clustering techniques. McKenzie et al. [14], for example, use error measures based on vorticity, divergence, and the vector gradient for a variational clustering approach. Their method tries to minimize the error of the value at all positions in a region to a corresponding average of the whole region (“proxy”) by adapting the partitioning producing the regions. An important difference of their gradient error measures to the one we use is that they compare the vector gradients at each position instead of comparing the vectors in the field directly to the local linear vector that is approximated. Du and Wang [2] present a distance measure for vectors that is mainly based on the direction of the vectors and use the measure to generate a vector field segmentation by centroidal Voronoi tessellation. What all these clustering methods have in common is that their regions do not adhere to a certain upper bound of the error but only try to minimize the error. Telea and van Wijk [30] use a distance function having ellipsoidal isocontours to determine the similarity of vectors for their clustering. The clustering technique introduced by Heckel et al. [6] is the only technique we know of that uses the distance of streamlines starting from a point as local error measure to compare a simplified vector field to the original vector field. Kuhn et al. [10] use a derived scalar field representing the bending energy of the vector field as input for the flow field clustering.

Very recently, Peng et al. [18] introduced an image-based vector field clustering approach. Their error measure includes cluster distance, velocity magnitude, direction, a local error, and the mesh resolution. Their clustering approach is designed for surfaces only, whereas we are also interested in three-dimensional vector fields. There are only few more vector field clustering methods in literature. However, describing them is beyond the scope of this paper because their methods of clustering are less related to our methods of linear neighborhood extraction. An overview discussing these methods has been given by Salzbrunn et al. [25].

3 LINEAR NEIGHBORHOODS

In the context of vector field visualization, the term *linear neighborhood* (LN) has been coined by Schneider et al. [26]. They introduce a *linear neighborhood* U_L to be the largest connected region around a vector field singularity $\mathbf{x}_c \in U_L(\mathbf{x}_c) \subseteq \mathbb{R}^3$ that is described by

$$U_L(\mathbf{x}_c) = \left\{ \mathbf{y} \in \mathbb{R}^3 \mid \frac{\|\mathbf{v}(\mathbf{y}) - J(\mathbf{x}_c) \cdot (\mathbf{y} - \mathbf{x}_c)\|}{\|\mathbf{v}(\mathbf{y})\|} < C_L \right\}, \quad (1)$$

where J denotes the Jacobian or vector gradient, \mathbf{v} the vector field, and $C_L \in \mathbb{R}$, $C_L > 0$ a certain bound. This equation is connected to the Taylor series. Effectively, it states that we are searching for areas with vanishing second and higher order derivatives. Therefore, the accepted error threshold C_L stands for the sum of all higher order terms of the vector field and should be as small as possible to describe a region that is dominated by a linear flow behavior.

In the following, we extend the notion of Schneider et al. to be applicable to any position in the domain and not only to singularities of the vector field. We call the extended concept *affine linear neighborhood* (ALN). Furthermore, we introduce a concept describing “how linear” the vector field is in a given region. We call it the *affine linear approximation error* (ALAE) with the corresponding *affine linear approximation* (ALA) of the region. The first method extracts linearly well approximable regions that are defined by a certain error threshold. The second method we introduce determines a certain error measure for any given region. In the end, both approaches provide a linearization of the field given by a system

of equations and the error of the approximation. The ideas behind these concepts will become clearer in the detailed descriptions following.

3.1 Affine Linear Neighborhood

Definition – We are using an extended definition of the linear neighborhood, which we call the *affine linear neighborhood* (ALN). The main difference, in contrast to the definition in Eq. (1), is that we define the ALN around an arbitrary seed position \mathbf{x}_s in the domain instead of only around singularities. To do so, we have to consider the actual flow vector at \mathbf{x}_s , i.e., through-flow, by using an affine linear vector field. Thus, the affine linear neighborhood U_{AL} for a seed position \mathbf{x}_s is defined as

$$U_{AL}(\mathbf{x}_s) = \left\{ \mathbf{y} \in \mathbb{R}^3 \mid \frac{\|\mathbf{v}(\mathbf{y}) - J_a(\mathbf{x}_s) \cdot (\mathbf{y} - \mathbf{x}_s)_a\|}{\|\mathbf{v}(\mathbf{y})\|} < C_L \right\}, \quad (2)$$

where, again, all positions \mathbf{y} , which by definition include \mathbf{x}_s , are simply connected. We use homogeneous coordinates for the calculations. Therefore, J_a denotes the Jacobian matrix augmented by $\mathbf{v}(\mathbf{x}_s)$ as its third or fourth column (depending on the dimension of the considered domain), $(\mathbf{y} - \mathbf{x}_s)_a$ indicates that the position $(\mathbf{y} - \mathbf{x}_s)$ is augmented by a 1, and the \mathbf{y} are chosen in a way that they form a connected region that includes \mathbf{x}_s . Similar to the linear neighborhood (1), ALN uses $(\mathbf{y} - \mathbf{x}_s)$ to transfer all positions into the frame of reference with origin \mathbf{x}_s but additionally incorporates the influence of $\mathbf{v}(\mathbf{x}_s)$ by using J_a instead of J , and the augmentation of the vector. Obviously, for vector field singularities at \mathbf{x}_s , i.e., $\mathbf{v}(\mathbf{x}_s) = \mathbf{0}$, this new neighborhood is the same as the traditional linear neighborhood around singularities as defined in Eq. (1).

Implementation – In the following, we introduce an algorithm to extract affine linear neighborhoods. This algorithm is designed for discrete datasets that are piecewise linear, respectively bilinear or trilinear, approximations of non-linear vector fields.

A straight-forward implementation of ALN would only be based on given data points. Nevertheless, the error function yields more complex results that we want to take into account. Before we are able to present an algorithm to compute the ALN in a piecewise trilinear vector field, we have to make some preliminary considerations.

We determine the boundary of the region U_{AL} by evaluating the equation

$$\frac{\|\mathbf{v}(\mathbf{y}) - J_a(\mathbf{x}_s) \cdot (\mathbf{y} - \mathbf{x}_s)_a\|}{\|\mathbf{v}(\mathbf{y})\|} = C_L. \quad (3)$$

Considering the three-dimensional case, the solution is a two-dimensional manifold representing the ALN boundary. Due to the equation’s complexity, an analytic computation is not feasible. Nevertheless, Eq. (3) allows us to easily determine all \mathbf{y} on linearly interpolated edges. This enables us to propose a construction of the manifold similar to the marching cubes algorithm [13]. Let \mathbf{y}_1 and \mathbf{y}_2 be the vertices of an edge $\mathbf{y}(\beta)$. Then, let $\beta \in [0, 1]$ the interpolation coefficient and $\mathbf{v}(\mathbf{y}) = \beta \mathbf{v}(\mathbf{y}_1) + (1 - \beta) \mathbf{v}(\mathbf{y}_2)$ the linearly interpolated vector field on this edge. When inserting this expression into Eq. (3) and solving for β , we obtain the quadratic equation

$$\begin{aligned} 0 = & \left[((\mathbf{v}(\mathbf{y}_1) - \mathbf{j}(\mathbf{y}_1)) - (\mathbf{v}(\mathbf{y}_2) - \mathbf{j}(\mathbf{y}_2)))^2 \right. \\ & \left. - C_L^2 (\mathbf{v}(\mathbf{y}_1) - \mathbf{v}(\mathbf{y}_2))^2 \right] \cdot \beta^2 \\ & + \left[(\mathbf{v}(\mathbf{y}_1) - \mathbf{j}(\mathbf{y}_1))(\mathbf{v}(\mathbf{y}_2) - \mathbf{j}(\mathbf{y}_2)) \right. \\ & \left. - (\mathbf{v}(\mathbf{y}_2) - \mathbf{j}(\mathbf{y}_2))^2 - C_L^2 (\mathbf{v}(\mathbf{y}_1) \mathbf{v}(\mathbf{y}_2) - \mathbf{v}(\mathbf{y}_2)^2) \right] \cdot 2\beta \\ & + \left[(\mathbf{v}(\mathbf{y}_2) - \mathbf{j}(\mathbf{y}_2))^2 - C_L^2 \mathbf{v}(\mathbf{y}_2)^2 \right], \end{aligned} \quad (4)$$

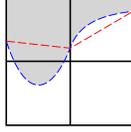


Figure 2: Linear approximation of a quadratic function in quadrilateral cells, where the light gray area represents the ALN. Red lines show the actually extracted boundary, by using the marching-cubes-based approach. Small structures arising from the quadratic function will be discarded and approximated by these borders.

where $\mathbf{j}(\mathbf{y}) = J_a \cdot (\mathbf{y} - \mathbf{x}_s)_a$ – see Appendix A in the additional material for the detailed derivation. This equation shows that the surface can be more complex than an isocontour in a linearly interpolated vector field because the scalar field defined here is quadratic. Unlike the marching cubes scheme, we can have zero, one, or two intersections on each line segment. However, we are only interested in edges with one intersection, because these edges provide the basis for the extraction of the linear approximation of the ALN boundary we will introduce below. Thus, we are consequently missing small structures resulting from the quadratic function that actually describes this boundary as shown in Fig. 2.

Our algorithm consists of two parts: In the first step, all inner cells of the linear neighborhood are extracted using a region growing approach. A pseudo code implementation is given in Algorithm 1. In the second step, a triangulated surface, representing the boundary of the affine linear neighborhood, is constructed.

The cell in which \mathbf{x}_s is located serves as starting point. If the cell is a simplex, the vertices of this cell can be described accurately by J_a and they are part of the ALN. Otherwise, only vertices that lie inside the ALN are of interest. By traversing all outgoing edges from each of these vertices and solving Eq. (4), it can be easily decided whether the edge intersects the affine linear neighborhood boundary or not. If the interpolation coefficient β has one unique solution in $[0, 1]$, one position on the boundary of the linear neighborhood was found. In this case, the endpoint of the currently tested edge will be marked as lying outside the linear neighborhood and the intersection point is stored together with the edge in a map. If no intersection was found, the endpoint of the outgoing edge belongs to the affine linear neighborhood and will be marked accordingly and all outgoing edges of this position will be marked for being checked next. If there are no more edges to be checked, the algorithm terminates and all positions lying inside the ALN have been found. We evaluate the approximated field at each position in the ALN using $J_a(\mathbf{x}_s)(\mathbf{y} - \mathbf{x}_s)_a$ and store the field to further investigate the approximated flow.

Due to the discrete nature of the input fields, we use a modified marching cells-based [16] isosurface construction algorithm to build a triangulated boundary surface. From the ALN region computation, we already know all cells that lie on the boundary of the volume. These are the cells we need for the triangulation. The marked cell vertices allow us to use the standard lookup tables from the marching cells algorithms to build a triangulated surface. Unlike marching cell-based algorithms, due to the quadratic equation, we can not determine the position of the intersection between the surface and the edge by using linear interpolation of the sampled error function. Therefore, we replace this linear interpolation by a look-up in the map of edge intersections, which we generated during the linear neighborhood volume extraction before. This allows us to obtain all intersections between the surface and the cell edges in constant time. We are aware of the fact that we may be missing a few smaller structures as depicted in Fig. 2, since a piecewise-linear approximation of the field is used. The diameter of these structures is always smaller than the length of an edge of a cell in the dataset.

Interpretation – In the previous section, We introduced an ap-

Algorithm 1: Compute affine linear neighborhood.

input: Seed position: seedPosition;
Linearity threshold: C_L
output: Positions inside ALN points: linearNeighborhood;
ALN boundary points: boundary

- 1: cell \leftarrow SearchCell(seedPosition)
- 2: startPositions \leftarrow GetCellVertices(cell)
- 3: **for all** pos \in startPositions **do**
- 4: **if** ComputeLinearity(pos) $\leq C_L$ **then**
- 5: positions \leftarrow {pos} \cup positions;
- 6: **while** HasNextElement(positions) **do**
- 7: pos \leftarrow NextElement(positions)
- 8: linearNeighborhood \leftarrow {pos} \cup linearNeighborhood
- 9: neighbors \leftarrow GetNeighboringPositions(pos)
- 10: **for all** neigh \in neighbors **do**
- 11: **if** EdgeNotChecked(pos, neigh) **then**
- 12: inter \leftarrow SearchIntersection(pos, neigh)
- 13: **if** inter **then**
- 14: boundary \leftarrow {inter} \cup boundary
- 15: **else**
- 16: positions \leftarrow {neigh} \cup positions

proach to find regions that can be safely described by a linear field approximation. Although, we will mainly refer to vortex core regions later on in the result section, of course our method is able to extract and depict all linearly describable flow phenomena.

In order to be able to choose an appropriate threshold C_L for the computation of the affine linear neighborhood, we now discuss how C_L influences the range of allowed magnitude difference and the angle α between the original and the linearly approximated vector in U_{AL} . To give an idea of the maximal possible variation of magnitude and angle, we rewrite Eq. 3 to

$$0 = R^2 - 2\cos(\alpha)R - C_L^2 + 1,$$

where α denotes the angle between the vectors $J_a(\mathbf{x}_s) \cdot (\mathbf{y} - \mathbf{x}_s)_a$ and $\mathbf{v}(\mathbf{y})$, and $R = \frac{\|J_a(\mathbf{x}_s) \cdot (\mathbf{y} - \mathbf{x}_s)_a\|}{\|\mathbf{v}(\mathbf{y})\|}$ is the length ratio of the approximated and the actual vector in the field – see Appendix B in the additional material for the detailed derivation.

To obtain a better impression on how the parameter C_L influences the deviations of vectors, we analyze the cases of angular variation at the same magnitude and the magnitude variation at the same vector direction independently. First, we set $\alpha = 0$ and solve for the ratio of vector magnitudes

$$R_{1,2} = 1 \pm \sqrt{1 + C_L^2 - 1} = 1 \pm C_L, \quad (5)$$

which defines the relation between the vector magnitudes as

$$(1 - C_L)\|\mathbf{v}(\mathbf{y})\| < \|J_a(\mathbf{x}_s) \cdot (\mathbf{y} - \mathbf{x}_s)_a\| < (1 + C_L)\|\mathbf{v}(\mathbf{y})\|.$$

Therefore, C_L defines the relative change of magnitude from the original vector. The second special case describes the maximal possible angle between $\mathbf{v}(\mathbf{y})$ and $J_a(\mathbf{x}_s) \cdot (\mathbf{y} - \mathbf{x}_s)_a$ in the case where they have the same magnitude. Setting $R = 1$, the maximum angle α_{max} can be estimated.

$$\begin{aligned} 0 &= 1 - 2\cos(\alpha_{max}) - C_L^2 + 1 \\ \alpha_{max} &= \arccos(1 - 0.5 C_L^2) \end{aligned} \quad (6)$$

We illustrate the possible magnitude and angle differences, described by the two equations above, in Fig. 3. The vectors accepted by our error measure, see Eq. (3), lie between the two extreme cases and no greater angle or magnitude differences are possible. To get a

better impression of this maximal possible angle between two vectors from U_{AL} we plotted α for an increasing relative error C_L in Fig. 4.

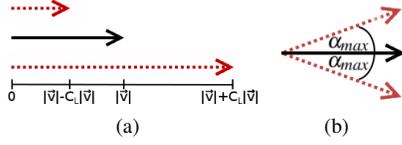


Figure 3: Black arrows represent vectors at the seed positions of ALNs and red arrows the vectors still accepted according to Eq. (3). Image (a) illustrates the possible magnitude differences between two parallel vectors as given by Eq. (5). Image (b) illustrates the possible angle differences between two vectors of the same length as given by Eq. (6).

3.2 Affine Linear Approximation

Affine-linear neighborhoods abstract from singularities by adding a through-flow component but still rely on dedicated points that act as seed point for the neighborhood and that dictate the linear behavior. The concept of *affine linear approximations* avoids the selection of these seed points by taking into account a sub-area or sub-volume of the field. Using this, we are determining the maximal or average error introduced by a possible linear approximation of the flow in a given region instead of extracting the region implied by a seed position and an error threshold.

Definition – An *affine linear approximation* of a field defined on a domain $D' \subseteq D \in \mathbb{R}^d$ in the field is the affine-linear map that approximates this field best. There are multiple ways of defining a “best fit”. For now, we derive the map by defining the three rows of the augmented Jacobian J_a using the system of j equations

$$PJ_a^i = V_i, \quad i \in \{1, 2, 3\}, \quad (7)$$

where P_{ji} is the matrix that contains the j points $(x_1^j, x_2^j, x_3^j, 1) = \mathbf{x}^j \in D'$ in each row, V_i the column vector with the i -th component of the flow vector $\mathbf{v}(\mathbf{x}^j)$ at position \mathbf{x}^j , and $J_a^i = (J_{ix}, J_{iy}, J_{iz}, J_{it})$ the i -th row of the augmented Jacobian matrix. As in the general case, this system is over-determined, we find an approximate solution by using the least-squares method [20]. As we are interested in an error per area or volume, in unstructured grids, we have to take the volumetric contribution of a value v into account by determining the covered volume. In linearly approximated data, the influence of a value defined at a vertex on the volume is determined by the volume of the neighboring cells, which we are using as weights to the equation. This leads to the modified equation for unstructured grids

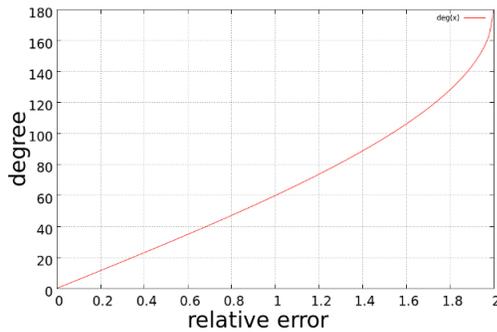


Figure 4: Plot of Eq. (6) illustrating the possible angular deviation α for a given C_L . Original and approximated vector have the same constant length.

$$P'J_a^i = V_i', \quad i \in \{1, 2, 3\}, \quad P' = WP, V_i' = WV_i', \quad (8)$$

where W is a diagonal matrix with $W_{ii} = \sqrt{A_i}$ and A_i is the sum of the areas or volumes of all cells neighboring vertex v_i .

To study the behavior, we compute the approximation on an n -ring neighborhood and call it *local level- n approximation* of the field. The quality of this approximation can be interpreted as a measure of local linearity of the field at approximation level n . Two values are of interest: The average error $ALAE_{avg}$, which is the weighted sum of all error values divided by the size of the region of interest, and the maximum error $ALAE_{max}$ within that region. The latter directly relates to the ALN as it provides the smallest error C_L that has to be used to extract at least this area using the ALN.

Implementation – To solve for J_a in Eq. (7), our implementation uses the singular value decomposition of the matrix P or P' to compute a pseudo-inverse in a memory- and time-efficient way. There are two ways of measuring the quality of fit. One is by using the null space of the Moore-Penrose inverse, which accounts for the distance metric used in the approximation. Instead, for an easier comparison with results from the previous sections, we employ the same error metric as used there as well.

Interpretation – The affine linear approximation provides a scale-based description of the dataset according to the local affine-linear behavior of the data. Thus, the affine linear approximation error (ALAE) can be seen as a measure of the local simplicity of the data at that approximation level and, combined with the ALA, locally provides an intuitive description of the data. Just as low ALAEs serve as an indicator of predominant affine-linear behavior in the region, higher ALAE values result where higher order vector fields occur. We would like to emphasize that the latter does not per se coincide with chaotic or turbulent flow. In addition, whereas $ALAE_{max}$ can only increase with increasing size of the approximated region, no such assumption can be made for the average error $ALAE_{avg}$. Implications thereof and examples are discussed in Section 5.

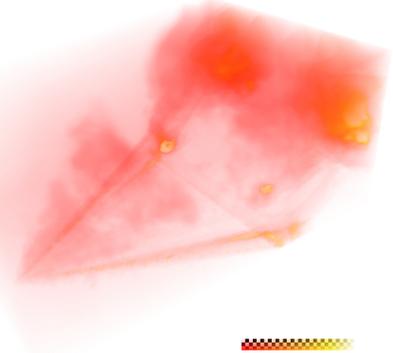


Figure 5: Direct volume rendering of the ALAE field at level two of the delta wing dataset. Its triangular shape is well perceptible. The areas with the largest ALAE, lie exactly behind the tail of the wing, can be clearly identified as the vortex breakdown bubbles.

3.3 Quality of Approximations

Both approaches presented here provide linear approximations of the given vector field in a certain region by approximating a suitable Jacobian matrix. We can deduce a specific equation that describes the entire flow in the corresponding region, which then allows analytical solutions of different quantities and visualization methods, such as the computation of streamlines, for instance. A crucial question is how far the approximated vector fields and the possibly contained flow features are corresponding to the flow features in

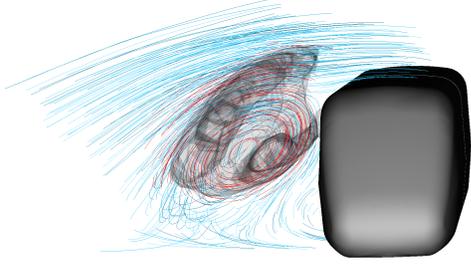


Figure 6: Front view of the ICE train with comparison of streamlines in an ALN next to the ICE. The ALN is shown in light gray. Red streamlines represent the linearly approximated vector field for $C_L = 0.2$ and blue streamlines the original vector field. The low error is achieved in the ALN, only. The swirling behavior of the approximated streamlines are well perceivable.

the original vector field. This question is partially answered by the particular user defined maximal error C_L or the computed ALAE. Additionally, one could determine an error measure as proposed by Heckel et al. [6], which is based on the deviation of streamlines that were started at the same position, but one in the approximated and one in the original vector field. We assume, that a study to validate detected flow features in the approximated vector field is reasonable but goes beyond the scope of this discussion, because we already inherently supply an error measure.

4 VISUALIZATION USING LINEAR NEIGHBORHOODS

The two different techniques that we introduced yield different types of data. The computation of the ALN results in a surface or line representing the boundary of the neighborhood, and a linear vector field in the ALN. A scalar field representing the ALAE for all positions in a dataset is the result of the affine linear approximation.

As the ALN represents a simplified vector field, rendering it using streamlines is a straightforward choice. A depiction of streamlines can for example be found in Fig. 6 where we compare streamlines in the approximated and in the original field. The streamlines in the approximated field are restricted to the ALN they represent and are drawn in red. Streamlines in the original field are blue and not restricted to the ALN. Fig. 6 shows that the affine-linear vector field matches the original field inside the ALN quite well.

The surface representing the boundary of the ALN can be rendered as a semi-transparent surface to allow looking into the ALN to be able to see the aforementioned streamlines in the ALN.

For the 3D scalar field representing ALAE, we employ isosurfacing to show the regions of particularly high ALAE (e.g., Fig. 11). For a complete overview of the different amounts of ALAE in the field, a direct volume rendering is better suited. See Fig. 5 for an example.

5 APPLICATION EXAMPLES AND DISCUSSION

In this section, we present and discuss the results of applying the ALN and ALAE computation to a number of analytic and CFD datasets.

5.1 Simple Analytic 2D Vector Fields

To provide first results for a better understanding of the ALN and to demonstrate the influence of the error threshold C_L on the set of positions of the linear neighborhood defined in Eq. (2), we created four synthetic two-dimensional datasets. They are analytic vector fields with different types of singularities centered at the origin of their domain. To add a non-linear component to the fields, they are multiplied with a Gaussian distribution given by $g(x, y) = \frac{1}{2\pi \cdot 0.1} e^{-\frac{1}{2} \cdot 0.1 \cdot (x^2 + y^2)}$. We therefore get fields with an increasing non-linear flow behavior with increasing distance from the

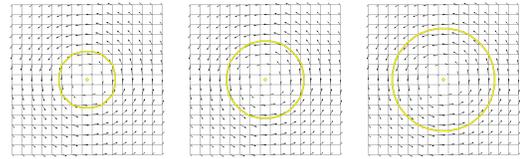


Figure 7: A synthetic two-dimensional vector field created by the multiplication of a Gauss kernel with a linear vector field, which contains a center point in its origin. There is a non-linear change in the flow behavior caused by the Gaussian kernel. This can be found by Algorithm (1). Its size depends on the chosen error threshold C_L . These images show the ALNs for $C_L = 0.2, 0.4$, and 0.8 , respectively.

origin. Every field was discretized, on a rectilinear grid with a resolution of 62×62 and a bounding box lying between $(-5.0, -5.0)$ and $(5.0, 5.0)$.

ALN – Because the vector fields are exhibiting a more and more non-linear behavior with increasing distance from the origin, there has to be a boundary of the linear neighborhood associated with the respective singularity. These boundaries are shown in Fig. 1. The size of these linear neighborhoods depends on the threshold C_L and on the Gaussian distribution. Exemplary computed neighborhoods for different thresholds applied to the center data set can be found in Fig. 7. The extracted affine linear neighborhood is growing with an increasing error threshold C_L , because larger thresholds allow a greater deviation from a perfect linear vector field.

5.2 Oseen Vortices



Figure 8: **Left:** Three ALNs seeded at the singularities of a data set containing two co-rotating Oseen vortices (resolution 500×500). **Center and right:** Color coding according to ALAE fields at approximation levels of four and eight, respectively. The relation of four, respectively eight, edge lengths to the grid size is indicated by the small red scales in the lower left corner of the ALAE fields. Red colored regions indicate non-linear flow behavior. The color map indicates that the ALAE values range from 0 (white) to 0.05 (dark red).

We discussed very simple cases in the previous section. To study the behavior of our new techniques in more detail, we are now looking at a more complex 2D vector field containing Oseen vortices. The Oseen vortex models a free vortex, i.e., the tangential velocity behaves inverse to the distance from the center. It decays due to viscosity. Given two co-rotational Oseen vortices, the flow field shows the interaction of a saddle and two center points. For a more detailed description of the model, we refer to a paper by Rom-Kedar et al. [21].

ALN – Because the flow is dominated by the interaction of the three singular points, we used them as seed points and determined the linear neighborhoods. The result is shown in Fig. 8 (left). The underlying line integral convolution (LIC) [28] shows the flow behavior. At the saddle point we can observe long “arms” extending to the center points. Here, the the flow can be described better by a linear vector field with respect to the error threshold. The linear neighborhoods of the center points are – in contrast to our synthetic example in 5.1 – clearly distorted. The flow induced by the saddle is influencing these two regions.

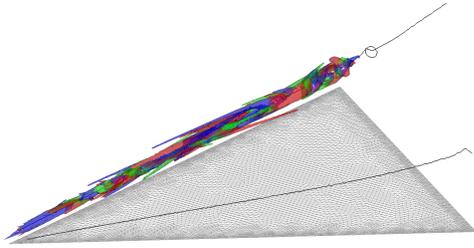


Figure 9: Depiction of the right primary vortex on the delta wing dataset as a composition of ALNs ($C_L = 0.1$) seeded equidistantly along the vortex core line. At this low error threshold, a composition of similar areas can be used to describe the shape of the vortex. Furthermore, one can very nicely perceive that the ALNs towards the vortex breakdown bubble shrink and twist because the flow behavior becomes more and more nonlinear (compare Fig. 11).

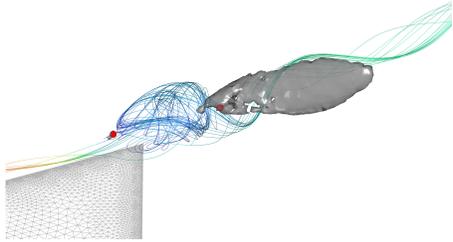


Figure 10: Affine linear neighborhoods at the singularities of the right vortex bubble formed behind the delta wing. Streamlines show the helical structure of the flow along the vortex core. This behavior remains stable until the flow enters the vortex bubble which is bounded by the vector field singularities (red spheres). The two gray surfaces depict the affine linear neighborhoods rooted at the singularities for $C_L = 0.1$.

Similar to [32], we are able to depict how different singularities influence each other. This means that, besides the possibility to use the extracted regions as basis for further investigations, they also allow us to get a deeper insight into the flow characteristics by directly looking at their structure.

ALAE – In Fig. (8) (center and right), we show a LIC image of the dataset with a coloring determined by two ALAE fields. The center image is computed with an n -ring neighborhood that has half the size of the neighborhood used for the right image. The size of regions with a high ALAE value are clearly increasing by doubling the used approximation level. This especially holds for the two strongest maxima that are located at the center points. It can also be seen that the high ALAE values coincide with the points where the saddle and the center ALN boundaries meet. Here, the flow cannot be described by a linear approximation. Besides, the area outside the vortices is nearly laminar and can be approximated linearly. Hence, the error decreases towards the image border.

5.3 Delta Wing

In the following, we examine a dataset resulting from a simulation of air flow around a single delta-type wing configuration. The dataset was computed in the context of numerical research into vortex breakdown [23]. The simulation was carried out for flight at subsonic speed. The initially already high angle of attack increases over time. The simulation shows the evolution of the primary, secondary, and tertiary vortex structures over time and the breakdown of the main vortices above the wing. The dataset is relatively large and contains about 12M cells. Although it contains multiple time steps, we limit our analysis to a single time step in this paper.

ALN – As the two primary vortices are the most important features in the dataset, we computed vortex core lines using the cell-based

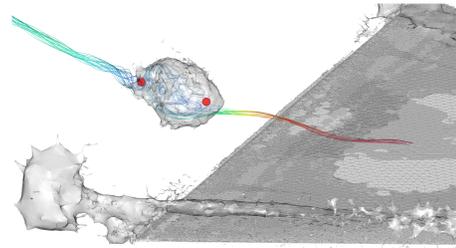


Figure 11: This image shows an $ALAE_{\max}$ isosurface at level two and $C_L = \sqrt{0.1}$ located above the tail of the wing, which is depicted by the grid. The vortex breakdown bubble is almost entirely enclosed by an area of high approximation error and, therefore, represents a region of non-linear behavior.

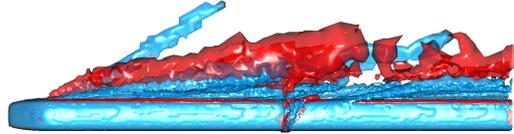


Figure 12: Top view of an ICE train for comparison between vortex core regions [8] illustrated by a $\lambda_2 = -10^{-9}$ isosurface in blue and a region with high non-linearity ALAE shown in red. The ALAE field was computed with parameters $C_L = \sqrt{0.0075}$ and $n = 2$. A region of high ALAE lies exactly between the two main vortices that are present on the lee side of the ICE train. High ALAE can also be observed where the flow separates from the top border of the train. The small structures in the middle of the train result from a small turbulence caused by the connector between the train and its first docked wagon.

vortex core line extraction method by Sujudi and Haimes [29]. Along the vortex core line of the right primary vortex, we equidistantly seed ALNs to visualize their local area of influence. We find areas around the vortex core line that are part of the linearly rotating motion. These ALNs are shown in Fig. 9. Speaking in terms of popular vortex models (e.g., the Rankine vortex [22]), we are hereby extracting the areas of solid body rotation along the vortex core line. Even though these areas align well around the vortex core line and define the expected tubular structure, an interesting irregularity can be seen in the “noses” and hollow tubes forming around the core line.

ALAE – The prevalent swirling flow pattern changes tremendously around the vortex breakdown bubbles forming behind the delta wing. Figures 10 and 11 show closeups of this behavior. We use streamlines and the two singularities that have been studied in detail by Rütten [23] and Garth et al. [4] before. The whole bubble, including the singularities, corresponds to a region of high ALAE, whereas the turbulent flow behind the bubble can be described well according to ALA, and larger ALNs can be extracted along the vortex core line (cf. Fig. 10). It is worth to note that the areas around both saddle points behave differently. Whereas the upstream saddle point is embedded in highly non-linear flow, the linear structure around the downstream saddle point serves well to describe parts of the downstream behavior, but the linear nature does not reach far into the vortex bubble.

5.4 ICE

A study of the track holding of the German high-speed train ICE led to this dataset. It is the result of a steady simulation of the train traveling at a velocity of about 250 km/h with wind blowing from the side at an angle of 30 degrees. The wind causes vortices to form and shed on the lee side of the train.

ALN – Fig. 6 shows an ALN seeded in the largest vortex based on low ALAE. At an approximation level of two, we are able to extract major parts of the main vortex and display the area of the helix structure that can be regarded approximately as a solid-body rotation around the vortex core line and rigid-body movement along the vortex core line. The figure also provides a comparison of streamlines in the linear field restricted to the cells containing the ALN and streamlines in the original field, which are not constrained to any area. Even though C_L is set to 0.2, which allows a greater variance between the approximated and original vectors, the comparison demonstrates the expected correspondence of the flow and nicely illustrates the helical structure.

This result shows that our method is able to extract similar regions as vortex extraction methods, that are based on the Rankine-vortex model for instance. That is namely, because the vortex core in such models is characterized by a linear behavior and we are exactly extracting these regions.

ALAE – Fig. 12 shows the dataset visualized using an isosurface of the popular vortex extraction criterion λ_2 [8] in comparison to an isosurface of $C_L = \sqrt{0.0075}$ in an $ALAE_{avg}$ field at level 2. The areas of high ALAE lie between the two main vortices and where the flow separates at the lee side from the top of the train. The extracted ALN in Fig. 6 shows a linearly rotating flow that corresponds with a part of the main vortices in Fig. 12. The region of high ALAE indicates a change in the flow, from linear to non-linear behavior and separates them from each other.

5.5 Tube

This dataset results from a simulation of the flow in a Francis turbine. The simulation was carried out by VA Tech Hydro. We are investigating a part of the simulation representing the unsteady flow in the draft tube (see Fig. 13) of the turbine. The flow is dominated by one moving main vortex. A detailed description and a picture of the turbine can be found in Appendix C in the additional material.

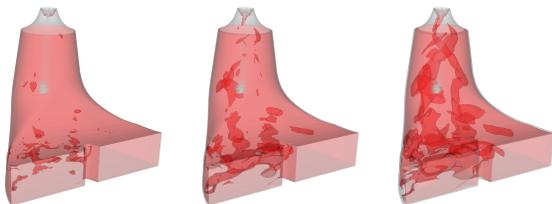


Figure 13: Comparison of ALAE in the turbine data set. The gray surface depicts the boundary of the draft tube. ALAE is shown using $C_L = \sqrt{0.1}$ isosurfaces at approximation levels two, four and six. The high ALAE values near the tube surface are probably caused by the shear in the boundary layer. With increasing level of approximation, more and more regions, which are only coarsely approximable by linear fields, arise at locations corresponding to vortices.

ALAE – Most isostructures indicating areas of high ALAE (Fig. 13), can be found close to the boundary of the dataset and in the highlighted regions in the inner part. The high error values at the boundary presumably occur due to increased shear forces that can be found in the outermost cell layer, only and are the same across all approximation levels. As the isosurface at $C_L = \sqrt{0.1}$ (Fig. 13) shows, the dataset can be approximated well locally (up to approximation level two), but at approximation level four, the first structures appear that cannot be approximated that well. At approximation level six, it can be clearly seen that the structures correspond to the main vortex and the secondary vortex that are rotating around a common center, see [27], and smaller regions close to the outlet that are known to have rotating flow. In comparison, Fig. 14 shows level four for different isovalues where the main vortex can be seen as an area that cannot be approximated at an error

Level	Tube	ICE	Delta Wing
2	1m 15s	1m 56s	7m 47s
4	5m 23s	11m 32s	56m 46s
6	14m 35s	42m 16s	3h 21m

Table 1: Computation times for the ALAE fields on different datasets at different neighborhood levels.

level below $C_L = \sqrt{0.05}$.

This is a good example to recall that we want to extract and investigate regions which are characterized by a linear flow and do not want to directly extract features like vortices. Especially in the case of vortices, we will only extract them, if the particular vortex model exhibits a linear behavior within the vortex core.

5.6 Computation Times

We implemented and executed our methods on *one* core of a workstation with 32 GB main memory and two Intel Xeon processors with four 2.4 GHz cores each. Both of our methods are perfectly suited for parallel execution. This will definitely be part of our future work.

ALN Fields – It is impracticable to state representative execution times of our ALN algorithm described in Section 3.1, because the results depend on the size of the region, which itself depends on the seed point \mathbf{x}_s , the threshold C_L , and the data. In order to give an impression, we can state that the computation and visualization of the ALNs around the eight singularities of the delta wing dataset for $C_L = 0.1$ took about 4 seconds. Two of the computed surfaces are shown in Fig. 10. In comparison, the computation of 935 ALNs at $C_L = 0.1$ seeded along the vortex core lines of the delta wing took 8 min 33 sec (approximately 550 ms per seed point), which includes building unstructured meshes for the surface and the datasets for the volume representation.

ALA and ALAE Fields – The computation times for the ALA fields can be found in Tab. 1. These times include the calculation of the ALAE fields with the averaged and maximum error and two fields representing the approximated Jacobian and augmented approximated Jacobian matrices for each position, depending on the chosen neighborhood level.

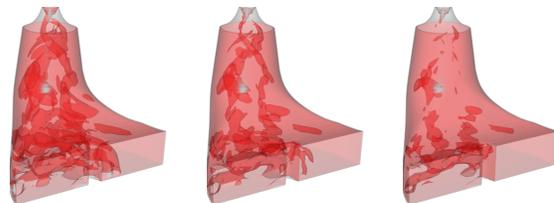


Figure 14: The same dataset as Fig. 13 compared at level four. Here, the isosurface is shown for C_L values of $\sqrt{0.025}$, $\sqrt{0.05}$, and $\sqrt{0.1}$. In agreement with the observations made in Fig. (13), structures indicating regions that are only coarsely approximable by linear fields are vanishing with increasing error bound.

6 CONCLUSION AND FUTURE WORK

We presented two approaches to investigate linearly describable parts of vector fields. One method to extract linear regions and a local measure to quantify the linearity in a n -ring neighborhood. We have shown that, by looking at linear parts of complex vector fields, regions containing different flow phenomena can be extracted and described by their linear approximation. As described in Section 5.2, this allows a deeper insight into the flow characteristics. Analyzing the introduced measure for the quality of local linear approximations (ALAE), enabled us to reveal new aspects of

the datasets as a whole and of certain features of the data in particular.

The presented measures and methods pave the way for many future research directions. An obvious next step surely is to provide a more efficient implementation of the presented methods. As mentioned, parallelization of the methods is easy and will be done first. After an advanced discussion about the choice of appropriate seed points of the affine linear neighborhood computation, the next step will be to quantify the flow behavior in these regions. This could be done by integrating characteristic quantities such as vorticity or velocity magnitude over the extracted regions and by providing a comprehensive depiction of the results. Furthermore, it is conceivable, that the extracted affine linear neighborhoods could serve as starting point not only for analytic solutions of flow quantities or streamlines but also for more complex calculations. In addition, it should be checked in how far our methods are complementary to each other, and to take ALAE fields into account in the course of the discussion of the choice of meaningful seed positions for the extraction of ALNs.

ACKNOWLEDGEMENTS

We thank Markus Rütten from German Aerospace Center (DLR) in Göttingen, VA Tech Hydro, and Ronald Peikert from ETH Zürich for providing the simulation datasets. Special thanks go to the *FAnToM* development group for providing the visualization software and the working environment. This work was partially supported by DFG grant SCHE 663/3-8 and by the European Social Fund (Application No. 100098251). Fig. 5 has been created using Amira - a system for advanced visual data analysis (amira.zib.de).

REFERENCES

- [1] W. C. de Leeuw and J. J. van Wijk. A probe for local flow field visualization. In *Proceedings of the 4th conference on Visualization '93*, pages 39–45, Washington, DC, USA, 1993. IEEE Computer Society.
- [2] Q. Du and X. Wang. Centroidal voronoi tessellation based algorithms for vector fields visualization and segmentation. In *Proceedings of the conference on Visualization '04, VIS '04*, pages 43–50, Washington, DC, USA, 2004. IEEE Computer Society.
- [3] C. Garth, F. Gerhard, X. Tricoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE TVCG*, 13(6):1464–1471, 2007.
- [4] C. Garth, X. Tricoche, and G. Scheuermann. Tracking of vector field singularities in unstructured 3D time-dependent datasets. In H. Rushmeier, G. Turk, and J. J. van Wijk, editors, *Proc. of the IEEE Visualization 2004*, pages 329–336. IEEE Computer Society, October 2004.
- [5] G. Haller. Lagrangian structures and the rate of strain in a partition of two-dimensional turbulence. *Physics of Fluids*, 13(11), 2001.
- [6] B. Heckel, G. Weber, B. Hamann, and K. I. Joy. Construction of vector field hierarchies. In *Proc. of the conf. on Visualization '99*, pages 19–25, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.
- [7] J. L. Helman and L. Hesslink. Representation and Display of Vector Field Topology in Fluid Flow Data Sets. *IEEE Computer*, 22(8):27–36, August 1989.
- [8] J. Jeong and F. Hussain. On the Identification of a Vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.
- [9] D. N. Kenwright, C. Henze, and C. Levit. Feature Extraction of Separation and Attachment Lines. *IEEE TVCG*, 5(2):135–144, 1999.
- [10] A. Kuhn, D. J. Lehmann, R. Gaststeiger, M. Neugebauer, B. Preim, and H. Theisel. A clustering-based visualization technique to emphasize meaningful regions of vector fields. In *Proc. of Vision, Modeling, and Visualization (VMV 2011)*, pages 191–198. Eurographics Association, 2011.
- [11] R. Laramée, H. Hauser, L. Zhao, and F. Post. Topology-based flow visualization, the state of the art. In *Topology-based Methods in Visualization*, Mathematics and Visualization, pages 1–19. Springer Berlin Heidelberg, 2007.
- [12] H. Löffelmann, H. Doleisch, and E. Gröller. Visualizing dynamical systems near critical points. In *Proceedings of the Spring Conference on Computer Graphics and its Applications 1998*, pages 175–184, Budmerice, Slovakia, April 1998.
- [13] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM.
- [14] A. McKenzie, S. V. Lombeyda, and M. Desbrun. Vector field analysis and visualization through variational clustering. In K. Brodlie, D. J. Duke, and K. I. Joy, editors, *EuroVis*, pages 29–35. Eurographics Association, 2005.
- [15] T. McLoughlin, R. S. Laramée, R. Peikert, F. H. Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum*, 29(6):1807–1829, 2010.
- [16] S.-N. J. G. K. Narayan, A. and B. Hamann. Isosurface extraction from hybrid unstructured grids containing pentahedral elements. In M. Kraus and Laramée, editors, *Proc. of Int. Conf. on Information Visualization Theory and Applications 2012 (IVAPP 2012)*, pages 660–669, 2012.
- [17] R. Peikert and M. Roth. The “Parallel Vectors” Operator - A Vector Field Visualization Primitive. In *Proceedings of the conference on Visualization '99: celebrating ten years*, pages 263–270, 1999.
- [18] Z. Peng, E. Grundy, R. S. Laramée, G. Chen, and N. Croft. Mesh-driven vector field clustering and visualization: An image-based approach. *IEEE TVCG*, 18(2):283–298, 2012.
- [19] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramée, and H. Doleisch. The State of the Art in Flow Visualization: Feature Extraction and Tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.
- [20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.
- [21] V. Rom-Kedar, A. Leonard, and S. Wiggins. An analytical study of transport, mixing and chaos in an unsteady vortical flow. *Journal of Fluid Mechanics*, 214:347–394, 1990.
- [22] M. Roth. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD thesis, ETH Zürich, 2000.
- [23] M. Rütten. *Topologische Untersuchung des Wirbelplatzens zur Identifikation von Wirbelplatzparametern*. PhD thesis, Helmut-Schmidt-Universität Hamburg, 2005.
- [24] F. Sadlo and R. Peikert. Efficient visualization of lagrangian coherent structures by filtered AMR ridge extraction. *IEEE TVCG*, 13(6):1456–1463, 2007.
- [25] T. Salzbrunn, H. Jänicke, T. Wischgoll, and G. Scheuermann. The state of the art in flow visualization: Partition-based techniques. In *Simulation and Visualization 2008*, pages 75–92. SCS Publishing House, February 2008.
- [26] D. Schneider, W. Reich, A. Wiebel, and G. Scheuermann. Topology aware stream surfaces. *Computer Graphics Forum*, 23(3):1153–1161, June 2010.
- [27] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. Interactive comparison of scalar fields based on largest contours with applications to flow visualization. *IEEE TVCG*, 14(6):1475–1482, 2008.
- [28] D. Stalling and H.-C. Hege. Fast and resolution-independent line integral convolution. In *Proceedings of SIGGRAPH '95*, pages 249 – 256, Los Angeles, California, 1995. Computer Graphics Annual Conference Series, ACM SIGGRAPH.
- [29] D. Sujudi and R. Haimes. Identification of Swirling Flow in 3D Vector Fields. Technical Report AIAA Paper 95–1715, American Institute of Aeronautics and Astronautics, 1995.
- [30] A. Telea and J. Van Wijk. Simplified representation of vector fields. In *Visualization '99. Proceedings*, pages 35 – 507, oct. 1999.
- [31] X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology Tracking for the Visualization of Time-Dependent Two-Dimensional Flows. *Computers & Graphics*, 26(2):249–257, 2002.
- [32] A. Wiebel, S. Koch, and G. Scheuermann. Glyphs for non-linear vector field singularities. In R. Peikert, H. Hauser, H. Carr, and R. Fuchs, editors, *Topological Methods in Visualization Workshop 2011*, pages 177–190. Springer, 2012.