

---

# Scale-Space Tracking of Critical Points in 3D Vector Fields

Thomas Klein and Thomas Ertl

Institute for Visualization and Interactive Systems (VIS), Universität Stuttgart  
{klein,ertl}@vis.uni-stuttgart.de

## Abstract

Scale-space techniques are very popular in image processing since they allow for the integrated analysis of image structure. The multi-scale approach enables one to distinguish between important features such as edges and small-scale features such as numerical artifacts or noise. In general, the same properties hold for vector fields such as flow data. Many flow features, e.g. vortices, can be observed on multiple scales of the data and also many features that can be detected are essentially artifacts of the employed interpolation scheme or originate from noise in the data. In this paper, we investigate an approach based on scale-space hierarchies of three-dimensional vector fields. Our main interest concerns how vector field singularities can be tracked over multiple spatial scales in order to assess the importance of a critical point to the overall behavior of the underlying flow field.

## 1 Introduction

The extraction of topological features has become a valuable tool for the analysis of vector field data that arise for example in computational or experimental fluid dynamics research. Vector field singularities in combination with a set of feature lines or surfaces, such as separatrices, provide the means of describing the qualitative behavior of a flow field in a strongly condensed form via the so-called topological skeleton. This enables one to cope with the ever growing dataset sizes in today's CFD applications.

Since typical datasets originating from simulations or real world measurements contain structures of different sizes or scales and different sets of features can be observed on certain ranges of scale, the notion of scale is an important concept in the analysis of the data. Therefore, an automatic or semi-automatic analysis tool for flow topology has to accommodate the inherent multi-scale nature of the underlying data.

Furthermore, many flow features, e.g. vortices, can be observed on multiple scales or resolutions of the data while many features that can be only detected at fine scales are essentially artifacts of the employed interpolation scheme or originate from noise in the data. Thus, a scale-aware technique can help to identify the overall structure of a given flow field and allows to distinguish the global structure from local effects, such as noise, and discretization or interpolation artifacts.

As in general the scales of the features are a priori not known, it seems to be reasonable to represent the data at multiple scales, successively eliminating fine-scale flow field structures. Nevertheless, in certain situations the scientist analyzing the data may be in possession of information that allows him to limit the parameter space to certain resolutions.

The concept of scale-space provides a well established framework to cope with these problems in a well defined way. Scale-space techniques have become very popular in image processing since they allow for the integrated analysis of the image structure. The multi-scale approach enables one to distinguish between important features, such as edges, and small-scale features, such as numerical artifacts or noise. By reason of this, scale-space techniques have been successfully applied to the automatic analysis of images in computer vision.

In this work we present our first attempt of a multi-scale topological analysis of flow data. We investigate an approach based on Gaussian scale-space hierarchies of three-dimensional vector fields. Our main interest concerns the tracking of important topological features, i.e. critical points, over multiple spatial scales in order to distinguish between local and global structures and behavior of the underlying vector field and numerical or noise-induced artifacts. This procedure is based on the assumption that fine scale structure and noise will be gradually eliminated on coarser scales while the dominating large scale flow structures will be persistent over multiple or all scales of the dataset. Thus, the scale length of the path or the number of scales over that a critical point can be tracked corresponds to the importance of this singularity to the overall topology of the flow.

## 2 Related Work

There is a huge body of literature dealing with the extraction and visualization of vector field topology. Since their first introduction to the context of visualization of two-dimensional vector field data by Helman and Hesselink [7, 8], topology-based methods have been established as one of the basic tools for flow analysis, and were soon generalized to three-dimensional fields, as well [8, 5].

Since then, many improvements and extensions have been published. The application of results from geometric calculus and Clifford algebra regarding vector field indices made it possible to detect and visualize also higher-order critical points and non-linear vector field topology. This has been investigated

in detail by Scheuermann et al. [18, 19, 20] for two-dimensional flows and by Mann and Rockwood [15] for three-dimensional vector fields.

As visualizations of the topology of three-dimensional flow fields typically involve a number of separating stream surfaces they suffer from occlusion problems and accordingly tend to get visually cluttered. To deal with that problem, Theisel et al. introduced the concept of saddle connectors [22] and boundary switch connectors [26] as a method for the simplified visualization of the topological skeleton of complex three-dimensional vector fields. However, the visual complexity, even of these simplified visualizations, depends heavily on the number of critical points involved.

To deal with large numbers of critical points and the resulting complexity of topological skeletons of vector fields the multi-scale nature of the data has been addressed in different ways. The simplest approach is to use a low pass filtering technique to suppress small-scale features and noise in the data and to do the extraction of the critical points on the filtered data. However, this approach cannot guarantee the invariance of the position and classification of critical points in the dataset. Therefore, a number of topological simplification methods have been proposed, e.g. the work by de Leeuw van Liere [2] or Tricoche et al. [23]. In contrast to the approach we will describe in the following, however, these methods often depend on the availability of the complete topological skeleton, not merely the critical points.

On the other hand, scale-aware feature extraction techniques for scalar valued data have a long tradition in image processing and computer vision. It was in the context of pattern recognition when the concept of scale of features and the scale-space paradigm emerged first [9]. Starting with the work of Witkin [28] and Koenderink [10] the concept of Gaussian scale-space representations has gained much attention in the image processing literature. The analysis of the so-called *deep structure* of images by means of an investigation of a multi-scale image representation, has become a valuable tool for feature detection and extraction in scalar images [14]. Of special interest in our case is the work by Lindeberg [13] and by Florack and Kuijper [3, 12] on the scale-space behavior of critical points in scalar fields, since the approach we will describe in Sec. 5 is closely related to their work on critical curves, i.e. the trajectories of singularities in the gradient field of a scale-space image. Although, we will disregard their extensive work on bifurcations and degenerate singularities based on the framework of catastrophe theory [11].

The tracking of features in vector fields has also attracted the interest of a number of researchers in flow visualization, especially in the case of the analysis of time-dependent vector fields. A general overview of the state of the art in feature extraction and tracking in flow fields can be found in Post et al. [17]. Particularly concerned with the tracking of vector field singularities as a topological feature in time-dependent two- and three-dimensional datasets are the works by Tricoche et. al [24, 25], Theisel and Seidel [21], and Garth et al. [4]. Feature tracking in scale-space representations of vector fields has been

investigated by Bauer and Peikert [1] for tracking the evolution of vortices in scale or time.

For the approach discussed in this work the papers by Theisel and Seidel [21] and Bauer and Peikert [1] are the two most relevant. In the first, the concept of feature flow fields is introduced. A vector field derived from the original flow field such that the evolution of the considered features is described by streamlines in this field. Thus, the feature tracking problem is reduced to simple streamline integration. We will come back to this technique in Sec. 5. In the second, vortex core lines based on the parallel vector operator are investigated. The construction of a linear scale-space representation of a flow field given on a three-dimensional unstructured grid is discussed and an algorithm for the tracking of the vortex core lines in scale-space based on a 4D surface extraction technique is presented. To our knowledge, this is the only work on scale-space feature tracking for flow visualization.

### 3 Scale-Space and Topology

#### 3.1 Gaussian Scale-Space

As already mentioned, scale-space techniques have become very popular in image processing since they allow for the integrated analysis of image structure at all scales. Here we will briefly outline the basic idea in case of a three-dimensional vector field.

In the following we will assume  $\mathbf{v} : M \rightarrow \mathbb{R}^3$  to be a vector field given on a three-dimensional domain  $M \subseteq \mathbb{R}^3$ . Introducing the scale parameter  $\tau > 0$ , the scale-space representation  $\boldsymbol{\nu} : M \times \mathbb{R}^+ \rightarrow \mathbb{R}^3$  of  $\mathbf{v}$  with  $\lim_{\tau \rightarrow 0} \boldsymbol{\nu}(\mathbf{x}, \tau) = \mathbf{v}(\mathbf{x})$  defines an embedding of the original vector field in a one-parameter family of derived vector fields, resulting in a hierarchy consisting of subsequently simplified or smoothed versions of the original data. Thus,  $\tau$  defines the scale-axis of the four-dimensional scale-space.

One possibility, and in fact the most often used, to generate such a one parameter family of vector field representations is the linear or Gaussian scale-space. Here the smoothing of the data is accomplished by the convolution  $\boldsymbol{\nu}(\mathbf{x}, \tau) = \mathbf{v}(\mathbf{x}) * g(\mathbf{x}, \tau)$  of the original signal with Gaussian filter kernels

$$g(\mathbf{x}, \tau) = \frac{1}{(2\pi\tau)^{3/2}} e^{-\frac{\|\mathbf{x}\|^2}{2\tau}} \quad (1)$$

of increasing standard deviation  $\sigma = \sqrt{\tau}$ . Which is equivalent to the solution of the linear diffusion or heat equation

$$\partial_\tau \boldsymbol{\nu}(\mathbf{x}, \tau) = \frac{1}{2} \Delta \boldsymbol{\nu}(\mathbf{x}, \tau), \quad (2)$$

with initial condition

$$\boldsymbol{\nu}(\mathbf{x}, 0) = \mathbf{v}(\mathbf{x}) .$$

Here  $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$  denotes the spatial Laplacian operator.

Hence, for a vector field given by data values on a discrete grid a Gaussian scale-space representation can be computed in at least two ways, by repeatedly filtering the data with sampled Gaussians of increasing variance and accordingly width or by solving the diffusion equation (2). Although, the second involves the solution of a partial differential equation, for large  $\tau$  this methods becomes increasingly efficient compared to using sampled Gaussians.

For the computation of the scale-space representation of our vector fields we have implemented both methods, the repeated convolution with separated Gaussians of increasing variance and accordingly increasing support and the iterative solution of the linear diffusion equation using a solver based on the SOR method. In both cases we work on three-dimensional Cartesian grids although the second approach could be very easily extended to unstructured tetrahedral meshes.

### 3.2 Scale-Space Topology

The additional degree of freedom introduced by the scale parameter, in fact, provides further topological features. There exist transitions between the different scale levels that cause topological changes. Subtle changes of the scale parameter can cause structural changes in the system's topology. These so-called bifurcations or catastrophes can occur in every dynamic system that depends on a set of varying control parameters, such as scale in our case. Thereby, the local topology changes from one stable state to another via a transient unstable state.

These phenomena have been studied extensively in the theory of dynamic systems and can be described in the framework of catastrophe theory that deals with how critical points of a parameter dependent dynamic system will evolve when the control parameters are continuously changed. We will not go into detail here, but refer to the literature [6]. An extensive account of the behavior of critical points of scalar fields, i.e. singularities of their gradient fields, under Gaussian blurring, i.e. in linear scale-space, can be found, e.g. , in the work by Florack and Kuijper [3, 12].

Many different types of possible bifurcations exist in a general dynamic system [6], but in flow topology one is mostly concerned with three general types of bifurcation events that can occur in the topology of parameter dependent vector fields: annihilations and creations of pairs of critical points and critical points changing their classification. The last, the so called Hopf bifurcation, describes for example the transition from a sink to a source or vice versa.

Unfortunately, the possibility of creation events in a linear scale-space representation seems to be contradictory to the goal of topology simplification. New (unstable) degenerate critical points can be created during the diffusion

process that subsequently will lead to the creation of a new pair of critical points. This is known as a static fold bifurcation. However, since we are only interested in the behavior of the critical points that exist in the original dataset, i.e. on the finest scale, we will in the following neglect this type of bifurcation event.

## 4 Detection of Vector Field Singularities

When talking about vector field singularities or critical points we will always assume spatial isolated zeros of the vector field with non-singular Jacobian matrix, i.e. first order critical points. This poses in general no problem, as line or surface singularities always correspond to degenerated critical points, i.e. points where the Jacobian of the field does not have full rank. Obviously, this also excludes higher-order critical points that may be present in non-linear vector fields [18].

The detection of vector field singularities is in general a numerical ill-posed and challenging problem. Noise and numerical inaccuracies will lead to false positives and actual singularities might be missed due to deficiencies of the employed interpolation scheme. As for all topology-based methods finding a complete seed set of critical points is a crucial part of the algorithm.

We have implemented and compared two approaches for the detection of critical points on a regular three-dimensional grid, namely the linearization of the vector field using a tetrahedral decomposition of the grid and a method based on the computation of winding numbers using geometric calculus. Both methods have certain strengths and weaknesses.

Decomposing the grid cells into tetrahedra corresponds to effectively linearizing the vector field, thus higher-order critical points—points with index not equal to 0 or  $\pm 1$ —cannot be detected directly by this method. Such points either are missed or show up as pairs of neighboring first-order critical points [18]. Although, for each grid cell multiple tetrahedra have to be processed, the actual computation per tetrahedron boils down to solving a linear equation system for the barycentric coordinates of the critical point’s position inside the cell. Therefore, this method is still the simplest and fastest way to compute vector field singularities as long as higher-order critical points can be neglected.

Second, we investigated the method described by Mann and Rockwood [15] that is based on computing the index of a critical point using winding numbers. At the moment this is regarded as the most general method, since it is not restricted to detecting first order critical points but can theoretically detect critical points of arbitrary index. Using results from geometric calculus Mann and Rockwood determine the index of a critical point by computing an integral over a closed surface. The result of this computation is always an integer value, that is either zero if no critical point is enclosed by the surface or equals the index of the critical point enclosed by the surface. Unfortunately,

there are also some problems with this approach. First of all, the value of the integral in general may not reflect the actual situation. Since the indices of multiple singularities enclosed by the surface add up in the result of the integration, an index of zero does not necessarily signify that no point is enclosed. The same applies for values not equal to zero. The problem is how to choose the appropriate spatial resolution for the enclosing manifold. At first sight one cell or the surface defined by its six faces seems to be a good choice, since this defines the maximal resolution of our data. But, as can be easily seen, a single cell is not sufficient to identify, for example, a dipole, at least if bilinear interpolation is used for sampling the cell faces. Thus, expensive higher-order interpolation schemes have to be used. But even when using tricubic interpolation, we experienced singularities that are missed. Furthermore, in order to achieve a reasonable accuracy for the integration, i.e. indices that are close to integer values, the sampling grid on the cell faces has to be sufficiently fine, which makes the computation even more expensive. Therefore, this method is not feasible for real datasets.

## 5 Tracking of Critical Points in Scale Space

In this section we will be concerned with the development of an algorithm for tracking non-degenerate singularities in scale-space. A straight forward solution would be to detect all critical points on all scales and then try to connect them according to some simple correspondence criteria, such as spatial distance or classification. There are a number of problems with this approach. In many situations such simple criteria are not sufficient to decide whether a critical point detected on a coarser scale really corresponds to a spatially close point on the next finer scale. Furthermore, tracking based on feature correspondence depends somehow or other on the definition of certain threshold values, e.g. for the maximum distance of two features considered to be equal. Often it is not possible to specify such values for an unknown dataset in advance. Thus, a fully automatic analysis of the data is not possible. In the following we describe two approaches that do not require such thresholds. The first is based on the concept of feature flow fields. The second exploits the explicit knowledge of the scale-space trajectory of a critical point provided by the implicit function theorem. Both approaches reduce the feature tracking problem to a stream line integration problem in a derived vector field, as will be described in the following.

### 5.1 Critical Point Tracking based on Feature Flow Fields

As introduced by Theisel and Seidel [21], a so-called feature flow field of a time-dependent three-dimensional vector field  $\mathbf{u}(\mathbf{x}, t) = (u_1(\mathbf{x}, t), u_2(\mathbf{x}, t), u_3(\mathbf{x}, t))^T$  is a four-dimensional vector field  $\mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, t), \dots, f_4(\mathbf{x}, t))^T$  derived from  $\mathbf{u}(\mathbf{x}, t)$  such that the time evolution of the considered features is described by

streamlines in  $\mathbf{f}(\mathbf{x}, t)$ . In the case of a critical point  $c$  this means each point on the streamline in  $\mathbf{f}(\mathbf{x}, t)$  starting from  $c$  is also a critical point, i.e. the value of  $\mathbf{u}(\mathbf{x}, t)$  must not change when traversing the streamline. Assuming a first order approximation of  $\mathbf{u}(\mathbf{x}, t)$  around  $c$  this implies that  $\mathbf{f}(\mathbf{x}, t) \perp \nabla u_i(\mathbf{x}, t)$  for  $i = 1, 2, 3$ . When we apply this to our scale-space representation  $\boldsymbol{\nu}(\mathbf{x}, t)$  by identifying time  $t$  with scale  $\tau$ , the scale-space feature flow field  $\mathbf{F}(\mathbf{x}, \tau)$  is given by

$$\mathbf{F}(\mathbf{x}, \tau) = \begin{pmatrix} -\det(\boldsymbol{\nu}_y(\mathbf{x}, \tau), \boldsymbol{\nu}_z(\mathbf{x}, \tau), \boldsymbol{\nu}_\tau(\mathbf{x}, \tau)) \\ \det(\boldsymbol{\nu}_z(\mathbf{x}, \tau), \boldsymbol{\nu}_\tau(\mathbf{x}, \tau), \boldsymbol{\nu}_x(\mathbf{x}, \tau)) \\ -\det(\boldsymbol{\nu}_\tau(\mathbf{x}, \tau), \boldsymbol{\nu}_x(\mathbf{x}, \tau), \boldsymbol{\nu}_y(\mathbf{x}, \tau)) \\ \det(\boldsymbol{\nu}_x(\mathbf{x}, \tau), \boldsymbol{\nu}_y(\mathbf{x}, \tau), \boldsymbol{\nu}_z(\mathbf{x}, \tau)) \end{pmatrix}, \quad (3)$$

where  $\boldsymbol{\nu}_x(\mathbf{x}, \tau)$ ,  $\boldsymbol{\nu}_y(\mathbf{x}, \tau)$ , and  $\boldsymbol{\nu}_z(\mathbf{x}, \tau)$  are given by the columns of the spatial Jacobian  $\boldsymbol{\nu}_\mathbf{x}(\mathbf{x}, \tau)$  of the scale-space representation.

Thus, the feature tracking problem is reduced to a four-dimensional streamline integration problem starting at an initial set of seed points given by the critical points encountered on the finest scale.

## 5.2 Singularity Tracking based on the Implicit Function Theorem

In this section, we will describe an alternative approach for critical point tracking that has been used, for example, by Lindeberg [13] for analyzing the behavior of local extrema in images under Gaussian blurring. Similar to the feature flow field approach, the basic idea here is also that using explicit knowledge about the actual trajectories of the critical points in scale-space can significantly improve the tracking results.

As we will show in the following, the evolution of a non-degenerate critical point in scale-space can be analyzed by means of the general implicit function theorem. Computing the trajectory of a critical point in scale-space can be regarded to be equivalent to finding the level set  $H : \boldsymbol{\nu}(\mathbf{x}, \tau) = \mathbf{0}$ . Then, for a given scale-space critical point  $(\mathbf{x}_0, \tau_0) \in H$  with non-singular matrix  $\boldsymbol{\nu}_\mathbf{x}(\mathbf{x}_0, \tau_0)$ , i.e.  $\det(\boldsymbol{\nu}_\mathbf{x}(\mathbf{x}_0, \tau_0)) \neq 0$ , the implicit function theorem states the following: In a local neighborhood of the critical point,  $\boldsymbol{\nu}(\mathbf{x}, \tau) = \mathbf{0}$  can be solved for  $\mathbf{x}$ . In other words, there exists a function  $\mathbf{h}(\tau)$  with  $\mathbf{x}_0 = \mathbf{h}(\tau_0)$  and  $\boldsymbol{\nu}(\mathbf{h}(\tau), \tau) = \mathbf{0}$ . Hence, the path of a critical point in the four-dimensional scale-space is a one-dimensional manifold, i.e. a curve. Although, it is not guaranteed that there exists an explicit representation of  $\mathbf{h}$ , the tangent of the curve in  $(\mathbf{x}_0, \tau_0)$  can be always computed as

$$\mathbf{h}'(\tau_0) = -(\boldsymbol{\nu}_\mathbf{x}(\mathbf{x}_0, \tau_0))^{-1} \boldsymbol{\nu}_\tau(\mathbf{x}_0, \tau_0). \quad (4)$$

Repeating this argument, the integration of the path of the singularity through scale-space can then be accomplished by solving the following differential equation

$$\partial_\tau \mathbf{h}(\tau) = -(\boldsymbol{\nu}_\mathbf{x}(\mathbf{h}(\tau), \tau))^{-1} \boldsymbol{\nu}_\tau(\mathbf{h}(\tau), \tau), \quad (5)$$

with initial condition  $\mathbf{h}(\tau_0) = \mathbf{x}_0$ .

### 5.3 Basic Tracking Algorithm

Regardless of the decision whether to use the method based on the feature flow field or the implicit function theorem in both cases a streamline has to be traced for all critical points detected in the original vector field. The methods differ only in that in the first case a four-dimensional streamline integration has to be performed whereas for the second method an integration in three dimensions is sufficient.

The basic algorithm then is performed as follows. We start by extracting all critical points on the finest scale level, i.e. the original dataset. Note that this does not guarantee a complete seed set for the topology of the scale-space representation of the data, but is sufficient to compute the scale-space evolution of the critical points under consideration. However, finding all critical points in a sampled vector field is a general problem in topological analysis.

Then for each critical point a streamline is traced based on the evaluation of either equation (3) or (5). Note also that in contrast to the more general case of streamline-based feature tracking in time-dependent flow fields investigated by Theisel and Seidel [21], it is sufficient to do only a forward integration of the streamline here.

In both cases scale-space derivatives have to be computed. This can be either accomplished by precomputing the field  $\nu$  for a fixed number of scales or by concurrent filtering/diffusion and stream line integration. The first is easy to implement but involves a high memory usage overhead for storing many three-dimensional vector fields. Since streamline integration is only forward in scale, only two scale levels are necessary when the computation of the scale-space representation and the stream line integration is done concurrently.

Furthermore, because we are working in a linear scale-space we have explicit knowledge about the derivative with respect to  $\tau$ . Since  $\partial_\tau \nu(\mathbf{x}, \tau) = \frac{1}{2} \Delta \nu(\mathbf{x}, \tau)$  the derivative with respect to scale can be derived from the second derivatives of  $\nu(\mathbf{x}, \tau)$  in space. Thus both  $\partial_{\mathbf{x}} \nu(\mathbf{x}, \tau)$  and  $\partial_\tau \nu(\mathbf{x}, \tau)$  can be computed by differentiation in the spatial dimension followed by a linear interpolation between two consecutive scale levels.

Last, the described streamline integration process has to be stopped as soon as an annihilation event is reached, as will be discussed in the next section.

### 5.4 Handling Bifurcations

Obviously, equations (3) and (5) are only valid as long as the Jacobian  $\nu_{\mathbf{x}}(\mathbf{x}, \tau)$  is not singular. Therefore we have to monitor the Jacobian matrix during the integration of the scale-space trace in order to capture the bifurcation events and stop the integration of the trace, accordingly. Note that the Jacobian  $\nu_{\mathbf{x}}(\mathbf{x}, \tau)$  is a smooth function of the scale parameter  $\tau$  and, accordingly, its determinant is also a smooth function of  $\tau$ . Thus, a sign change of the Jacobian determinant signifies a bifurcation. There are now two possibilities for the

kind of event that has occurred. The critical point is either passing through the unstable state of a Hopf-type bifurcation or a fold bifurcation point has been reached and the traced critical point is annihilated. In the second case, the condition is sufficient to detect the event since in an annihilation event always two critical points with opposite sign of their Jacobian determinant are involved. But although a wide range of possible Hopf-type bifurcations can be also detected that way, some of them are hard to detect since the Jacobian determinant does not change its sign but instead passes through a second order zero of the characteristic polynomial. In this case it is very unlikely, that the event would be even noticed. Nevertheless, this problem could be solved by computing the eigenvalues of the Jacobian. However, this poses a large computational overhead that is in fact completely unnecessary. Since we are interested in classifying the critical points according to their scale-space lifetime Hopf bifurcations can be safely neglected because only the type of the critical point is changed not the fact of its existence. That means, the problem is not whether a Hopf bifurcation has been missed, but in the case a zero determinant is encountered if it is an annihilation event or not. Therefore, if  $\det(\nu_{\mathbf{x}}(\mathbf{x}, \tau))$  is zero we have to check if it is a real zero crossing by going one step further in the direction of the streamline before terminating the streamline integration.

### 5.5 Predictor-Corrector Tracking

Unfortunately, both methods turned out to be numerically unreliable. Although, they work well for simple generated test datasets both fail to capture the correct behavior for noisy real world datasets. Even when we use a fourth order Runge-Kutta integration scheme with adaptive step size for the computation of the streamlines, the computed traces did very soon deviate from the actual path of the critical points in scale-space that have been computed for comparison by repeated extraction of critical points on multiple scale levels. The major problem seems to be the accuracy of computed derivatives. They have to be numerically approximated from the sampled vector field and, although we use a tricubic interpolation scheme, their accuracy, in particular of the second derivatives, is not very high. Since using even higher order interpolation would give rise to a disproportionately large computational effort this is not an option. Therefore, we have combined the scale-space tracing with a Newton-Raphson method in a predictor-corrector approach. In each step of the streamline integration a prediction is computed by integrating in the direction indicated by either the feature flow field  $\mathbf{F}(\mathbf{x}, \tau)$  or the right hand side of equation (5). Afterwards, this prediction is refined using a typically very small number of Newton-Raphson iterations. Now it turned out that even a simple third order Runge-Kutta method is sufficient to compute reliable scale-space traces.

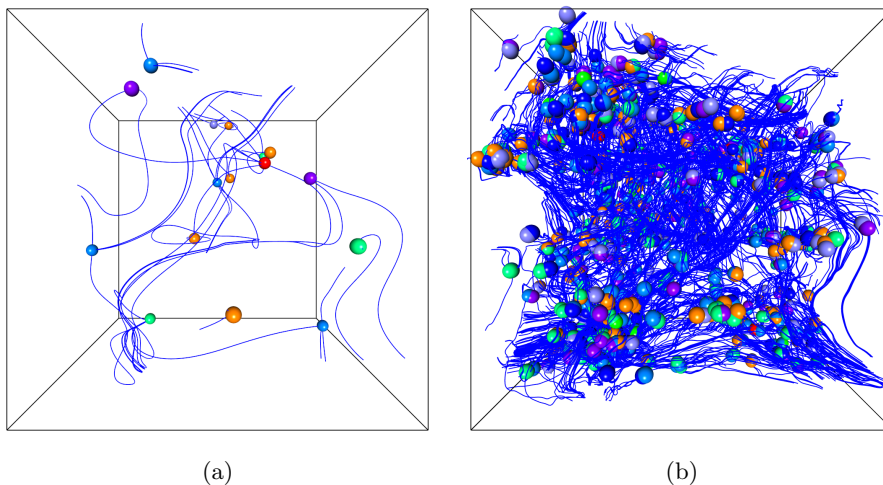
## 6 Results

In this section we will present results of the above described scale-space tracking approach applied both to a generated test dataset as well as to a real flow dataset.

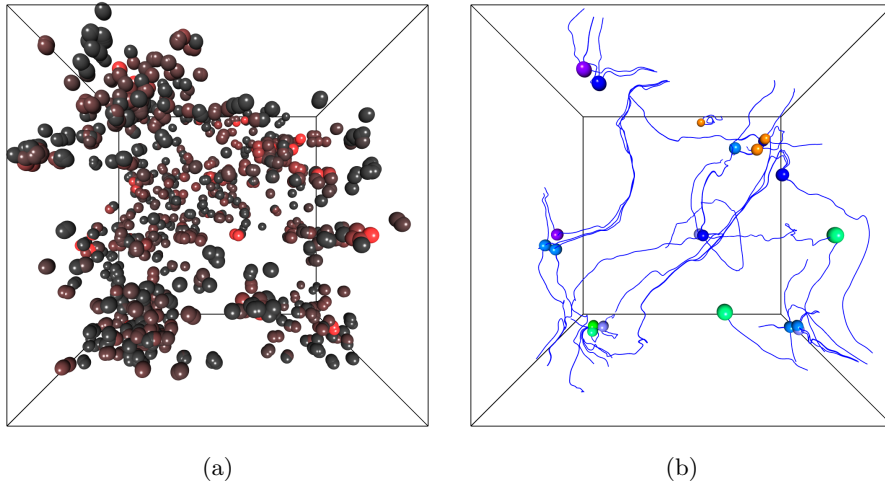
Our first dataset is an artificial test dataset. It was created by resampling a random generated  $10^3$  vector field to a  $50^3$  grid using tricubic filtering. In this dataset 16 first-order critical points (3 saddles, 3 focus saddles and 10 foci) have been detected. The original 16 critical points in combination with streamlines seeded in their vicinity are shown in Fig. 1a. The same dataset after adding some noise is shown in Fig. 1b. In this case, normal-distributed noise was added to 20% of the vector components of the field, which leads to a rather high signal-to-noise ratio of approximately 11dB for this dataset.

The number of critical points that can be detected now is 1307 and the topology is much too complex to be of any practical use. Applying the proposed scale-space tracking scheme, enables us to distinguish between critical points that have been solely introduced due to noise and critical points that represent the dominating flow behavior.

For this test, the points were traced using the scheme derived in Sec. 5.2 over the scales from  $\tau = 0$  to  $\tau = 1.5$ . The computation took approximately 70 second on a machine equipped with an AMD Opteron 2.0GHz processor and 8GB of main memory. 8.5 seconds were spent for computing the scale-space representation and 61 second for the actual streamline integration. The numbers for the feature flow field based method of Sec 5.1 are comparable and identical results are produced for both methods. Fig. 2a shows the result of



**Fig. 1.** A random generated test dataset. Critical points are color coded according to their classification. Foci are shown in red and green hues, while saddles are colored in blue tones. (a) Original data. (b) Data with noise added.

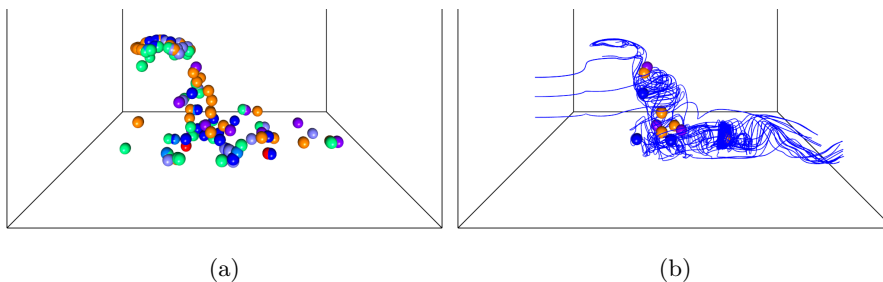


**Fig. 2.** The same test dataset as shown in Fig. 1. (a) Scale-space lifetime of the critical points in the interval  $\tau = 0 \dots 1.5$  computed by our algorithm. Red color indicates stable critical points, while gray colored points are very short-lived. (b) Critical points filtered by their lifetime. Only points that persist at  $\tau_{\min} = 1.0$  are shown.

this computation. A gray to red color-coding is applied to indicate the scale-space lifetime of the critical points. Gray points are very short lived while bright red points could be traced over the whole scale interval. The actual scale-space traces of the critical points are not shown in the images, since they do not provide much of additional information. They may only serve to identify pairs of critical points that participate in annihilation events.

Last, in Fig. 2b the result of filtering the points according to their scale-space lifetime is shown. Only points that have a lifetime larger than  $\tau_{\min} = 1.0$  are shown. Note that the streamlines have been integrated starting from the remaining points in the original noisy field not in a smoothed representation. The same holds for the classification of the critical points. Of course, it is not possible to recover the original topology of the noise-free dataset but the overall behavior is much more apparent.

As a second example we present the application of the scale-space tracking to a real CFD dataset—a simulation of the flow past a circular cylinder. There are 141 critical points that can be detected in this dataset which account for the complex flow topology in the wake behind the cylinder. Fig. 3a shows the critical points detected in the flow field. Only the lower half of the  $180^3$  data set is shown in this image since there were no critical points detected in the upper part of the dataset. For both images of Fig. 3 the same type of color-coding as for those in Fig. 1 has been used. Computing the scale-space lifetime of the critical points for this dataset took approximately 4:58 minutes using the same machine as for the first example. Since this dataset



**Fig. 3.** The flow behind a spherical cylinder. The same kind of color coding is used as for Fig 1. (a) Critical points on the fines scale, i.e. the original flow field. (b) Scale-space lifetime of the critical points in the interval  $\tau = 0 \dots 10$  is computed by our algorithm. Then the Critical points were filtered by their lifetime ( $\tau_{\min} = 6.3$ ) and the remaining points have been used to seed stream lines in their vicinity.

is significantly larger than the artificial one used in the previous example and also contains far less critical points, most of the time (4:34 minutes) was spent for computing the scale-space flow field representation while for the relatively small number of critical points the scale-space tracking could be done in only 24 seconds. Seeding streamlines only around critical points that can be tracked for  $\tau \geq 6.3$  does significantly reduce the amount of streamlines that are displayed and accordingly the problems with visual clutter, but yet one can still clearly discern the overall behavior of the flow. Again, both tracking approaches produce comparable results.

## 7 Conclusion and Future Work

In this paper we have presented a first approach on a tracking algorithm for vector field singularities in scale-space that uses explicit knowledge of the evolution of the field along the scale-axis. Our approach is based on streamline integration in a derived vector field that allows us to track the evolution of first-order critical points in linear scale-space. Both a feature flow field or the gradient of the level-set implicitly defined by the scale-space curve of a critical point have been used to define that field. Furthermore, we proposed a predictor-corrector method to deal with numerical problems that arise when computing vector field derivatives in a noisy dataset. Last, results have been shown for both artificial test datasets and real CFD data.

Although, the use of the linear scale-space might not be the final answer for building a scale representation of a flow field, we, nevertheless, think that scale-space techniques are a promising way to deal with noisy and highly complex flow datasets. However, a rigorous analysis of the behavior of critical points of 3D vector fields under Gaussian blurring, similar to the work on scalar fields by Kuijper [11], has still to be done.

In future work we want to extend our current implementation to use unstructured grids and other multi-scale analysis schemes. Especially the use of wavelet based smoothing seems to be a promising way to overcome the inherent problem that the linear scale-space representation does not necessarily imply topology simplification. As other possible extensions, we like to generalize the scale-space tracking algorithm also to unsteady flows, i.e. to track the singularities not only over scale but simultaneously over time. This involves the problem of how to compute and interpret the time-surfaces of critical curves in a five-dimensional space. However, this problem is quite similar to the problem of tracking of other extended vector field features, for example vortices. Another general issue in topological analysis based on critical points is the conceptual problem of critical points to be not Galileian-invariant. That means important features can be missed. Therefore we want to look into recent approaches for the treatment of the Galileian-invariance problem [16, 27] in order to guarantee a complete seed point set for the tracking.

### Acknowledgments

This work was funded by the German Research Council (DFG) as part of project SFB 382.

### References

1. D. Bauer and R. Peikert. Vortex tracking in scale-space. In *Proceedings of the Symposium on Data Visualisation '02*, pages 233–240, 2002.
2. W. de Leeuw and R. van Liere. Visualization of global flow structures using multiple levels of topology. In *Proceedings of the Symposium on Data Visualisation '99*, pages 45–52, 1999.
3. L. Florack and A. Kuijper. The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision*, 12(1):65–79, 2000.
4. C. Garth, X. Tricoche, and G. Scheuermann. Tracking of vector field singularities in unstructured 3d time-dependent datasets. In *Proceedings of IEEE Visualization '04*, pages 329–336, 2004.
5. A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Proceedings of IEEE Visualization '91*, pages 33–40, 1991.
6. J. Guckenheimer and P. Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. Springer Verlag, 1986.
7. J. L. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, 22(8):27–36, 1989.
8. J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Comput. Graph. Appl.*, 11(3):36–46, 1991.
9. T. Iijima. Basic theory on normalization of a pattern (in case of typical one-dimensional pattern). *Bulletin of Electrical Laboratory*, 26:368–388, 1962.
10. J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

11. A. Kuijper. *The Deep Structure of Gaussian Scale Space Images*. PhD thesis, Utrecht University, 2002.
12. A. Kuijper and L. Florack. Calculations on critical points under gaussian blurring. In *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision '99*, 1999.
13. T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
14. T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, 1998.
15. S. Mann and A. Rockwood. Computing singularities of 3d vector fields with geometric algebra. In *Proceedings of IEEE Visualization '02*, pages 283–290, 2002.
16. K. Polthier and E. Preuß. Identifying vector field singularities using a discrete Hodge decomposition. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 113–134. Springer-Verlag, 2003.
17. F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramée, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.
18. G. Scheuermann, H. Hagen, H. Krüger, M. Menzel, and A. Rockwood. Visualization of higher order singularities in vector fields. In *Proceedings of IEEE Visualization '97*, pages 67–74, 1997.
19. G. Scheuermann, H. Krüger, M. Menzel, and A. P. Rockwood. Visualizing non-linear vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, 1998.
20. G. Scheuermann, X. Tricoche, and H. Hagen. C1-interpolation for vector field topology visualization. In *Proceedings of IEEE Visualization '99*, 1999.
21. H. Theisel and H.-P. Seidel. Feature flow fields. In *Proceedings of the Symposium on Data Visualisation 2003*, pages 141–148, 2003.
22. H. Theisel, T. Weinkauff, H.-C. Hege, and H.-P. Seidel. Saddle connectors - an approach to visualizing the topological skeleton of complex 3d vector fields. In *Proceedings of IEEE Visualization '03*, page 30, 2003.
23. X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proceedings of IEEE Visualization '01*, pages 159–166, 2001.
24. X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology-based visualization of time-dependent 2d vector fields. In *Proceedings of the Symposium on Data Visualisation '01*, pages 117–126, 2001.
25. X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computer & Graphics*, 26(2):249–257, 2002.
26. T. Weinkauff, H. Theisel, H.-C. Hege, and H.-P. Seidel. Boundary switch connectors for topological visualization of complex 3d vector fields. In O. Deussen, C. Hansen, D. A. Keim, and D. Saupe, editors, *Proc. Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '04)*, pages 183–192, 2004.
27. A. Wiebel, C. Garth, and G. Scheuermann. Localized flow analysis of 2d and 3d vector fields. In *Proceedings of Eurographics / IEEE VGTC Symposium on Visualization '05*, pages 143–150, 2005.
28. A. P. Witkin. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 1019–1022, 1983.