

Towards Automatic Feature-based Visualization

Heike Jänicke¹ and Gerik Scheuermann¹

1 Image and Signal Processing Group

University of Leipzig

Leipzig, Germany

{jaenicke,scheuermann}@informatik.uni-leipzig.de

Abstract

Visualizations are well suited to communicate large amounts of complex data. With increasing resolution in the spatial and temporal domain simple imaging techniques meet their limits, as it is quite difficult to display multiple variables in 3D or analyze long video sequences. Feature detection techniques reduce the data-set to the essential structures and allow for a highly abstracted representation of the data. However, current feature detection algorithms commonly rely on a detailed description of each individual feature. In this paper, we present a feature-based visualization technique that is solely based on the data. Using concepts from computational mechanics and information theory, a measure, local statistical complexity, is defined that extracts distinctive structures in the data-set. Local statistical complexity assigns each position in the (multivariate) data-set a scalar value indicating regions with extraordinary behavior. Local structures with high local statistical complexity form the features of the data-set. Volume-rendering and iso-surfacing are used to visualize the automatically extracted features of the data-set. To illustrate the ability of the technique, we use examples from diffusion, and flow simulations in two and three dimensions.

1998 ACM Subject Classification I.3.6 Methodology and Techniques, I.3.7 Three-Dimensional Graphics and Realism

Keywords and phrases Feature Detection Techniques, Feature-based Visualization, Local Statistical Complexity

Digital Object Identifier 10.4230/DFU.SciViz.2010.62

1 Introduction

When analyzing their data-sets, one of the important questions of researchers is: Did I see everything that is relevant? Usually the domain experts can name several structures that they are interested in and that have significant influence on the system's evolution. Commonly, such structures are called features. In fluid dynamics, for example, they comprise structures like vortices, separation and attachment lines, cycles, and stagnation points. Detecting and visualizing these structures automatically is of great help for the domain experts. They get a simplified description of the system and can immediately understand basic properties of their data-set.

In order to detect these prominent structures in a data-set automatically, mathematical descriptions are required. Some features like stagnation points can be detected very easily, as they are simply zeros in the vector field. Other features like vortices, however, are very hard to define mathematically. Several different detection methods based on vorticity, λ_2 , or the Sujudi and Haines algorithm exist, but neither is capable of detecting vortices in all scenarios (Galilean invariance). The vortex example illustrates that more complex features are often hard to describe with a simple algorithm or formula, which gets even tougher in an unsteady setting. Here, a less restrictive feature definition would be beneficial.



© H. Jänicke and G. Scheuermann;
licensed under Creative Commons License NC-ND

Scientific Visualization: Advanced Concepts.

Editor: Hans Hagen; pp. 62–77



DAGSTUHL
FOLLOW-UPS

Dagstuhl Publishing

Schloss Dagstuhl – Leibniz Center for Informatics (Germany)

A second problem that arises when looking for features is the fact that there is no general definition of a feature. In general, features are phenomena, structures or objects in a data set of interest for the underlying problem [19]. Thus, features strongly depend on the application and the user. Users from computational fluid dynamics (CFD), magnetic resonance imaging (MRI) and biological system simulation will be looking for different features and for each field a different set of tools is required to detect the structures the domain experts are interested in.

Even when provided with the appropriate set of tools, the user still has to run several algorithms to detect all the different features and usually has to specify parameters for each of them. Hence, the user has to start five to ten algorithms, set parameters, wait for the results, check whether something has been found, verify the results and look for structures that are not included in the list of standard features. Doing this entire procedure for several data-sets can become quite wearisome and much easier feature detection process would be desirable.

Summarizing these last three scenarios, we found the following weak points of the standard feature detection procedure:

- Most features can be found without domain knowledge even by a novice. Why can't computers do this?
- A feature may depend on the application. User dependence sounds weird in natural sciences (or engineering).
- If the data describes a physical simulation, a feature should depend only on the data.

To deal with these problems, the feature detection procedure described in the last scenario has to be highly simplified. The algorithm for the identification of relevant structures we think of, should look something like this:

1. Load the simulation data.
2. Run the feature detection algorithm.
3. Get a visualization with highlighted features (i.e. the most important regions).

Moreover, we want the logic behind the algorithm to be easy to understand and that the algorithm does not need a definition or name for all the different types of features it detects. The second requirement ensures that new structures can be found that have not been identified as features before.

Hence, the goal of the paper is to present a new way towards a feature-based visualization that does not need a priori definitions of structures that are considered relevant. On the contrary, relevance is to be directly defined by the data itself and the user is presented those structures that differ from the basic patterns in the data-set, i.e., the features of the data-set.

2 Related work

Much work has been done in the field of feature detection and visualization. In general four different concepts can be distinguished: image processing, topological analysis, physical characteristics, and partition-based approaches. *Image processing techniques*, e.g. Ebling et al. [6], Schlemmer et al. [24] and Heiberg et al. [9], often apply pattern matching approaches. Here a two or three-dimensional pattern is predefined and similar structures are found in the data-set. Although these techniques are very flexible with respect to finding certain patterns with different scale and/or orientation, the user still has to define a sample pattern as reference for each of the structures he/she is looking for. *Topological analysis* clusters regions of similar behavior/structure. Examples in this area can be found in the survey by Scheuermann et al. [23]. The topological analysis of a data-set provides an automatic simplification. However, there is no classification of the importance of the different structures that have been identified. Many feature detection methods are based on the analysis of *physical characteristics* (e.g. Garth et al. [7], Roth [20]), as these are the most intuitive descriptions for domain experts. Though many excellent methods fall into this category, they all have the problem

that they are very restrictive concerning the definition. A detailed description of flow feature detection techniques that fall into these three categories can be found in the survey by Post et al. [19]. The idea behind partition-based approaches is to separate the domain into regions of similar structure or behavior. Streamline predicates [22] and pathline attributes [28], for example, cluster integral lines in the data-set with given properties. The method we are going to present falls into this category, as it partitions the domain into areas that feature distinct structures and those that do not. Partitioned-based approaches are summarized in the paper by Salzbrunn et al. [21].

As mentioned before, these standard feature detection methods commonly rely on a given definition or description of the feature to be found. What we are looking for is a feature description given by the data-set itself. The third step of the algorithm we have in mind (Section 1) already reveals the direction we are aiming at. We do not want to provide exact feature definitions, but are looking for regions of high importance in the data-set. Important is to be understood in an information-theoretic way, i.e. we want to identify the regions with the highest information content or complexity.

In the literature, a large variety of complexity measures are available, e.g., [2, 4, 8, 16, 27]. Common measures originating from the analysis of strings of data are Shannon entropy [27] and algorithmic information [1]. Shannon entropy is a measure of the uncertainty associated with a random variable, whereas the algorithmic information is roughly speaking the length of the shortest program capable of generating a certain string. Both measures have in common that they are measures of randomness. In complex systems however, randomness is commonly not considered to be complex. Likewise, Hogg and Huberman [11] state that complexity is small for completely ordered and completely disordered patterns and reaches a maximum inbetween. A different approach was taken by Grassberger [8], who defined complexity as the minimal information that would have to be stored for optimal predictions. Based on this idea, statistical complexity [4] was introduced identifying the complexity of a system with the amount of information needed to specify its causal states, i.e., its classes of identical behavior. In order to analyze random fields, a point-by-point version was formulated by Shalizi [25] called local statistical complexity.

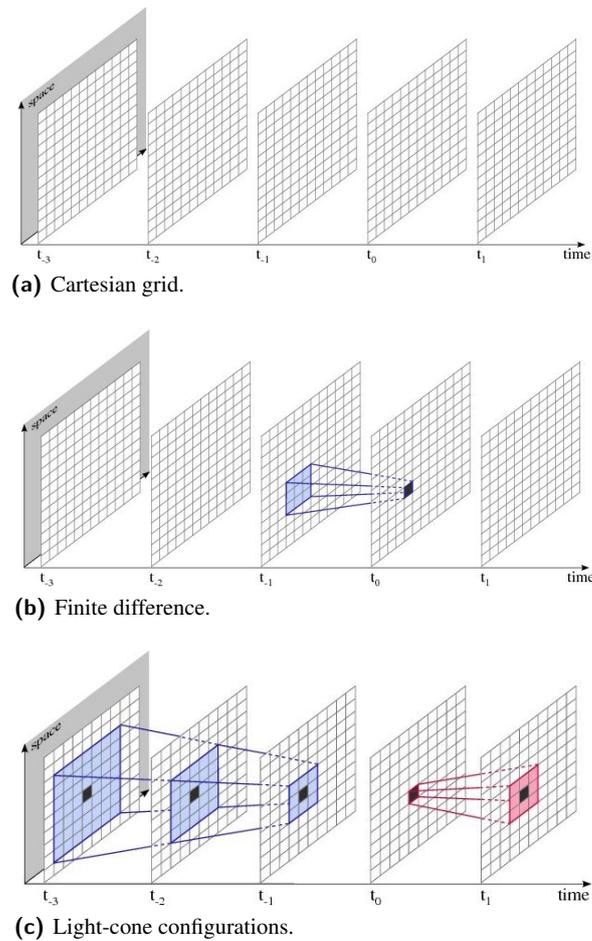
3 Specifications

The following work is based on the ideas by Shalizi et al. [26], which assumes the following properties of a data-set:

1. The data stems from a PDE simulation (in engineering or natural sciences).
2. The solver is based on a finite difference scheme on a Cartesian grid.
3. The data is unsteady and all time-steps and independent variables are available.

The first requirement ensures that the process creating the data is the same at each position in the resulting field. As PDEs are the standard definition of physical systems, e.g. the Navier-Stokes equations for fluid flow, this demand sets no limitations. The second requirement allows for the comparison of local neighborhoods. A sample Cartesian grid is given in Figure 1a. Using finite difference schemes as solver, clearly defines a local neighborhood that is used to compute the value in the next time-step as illustrated in Figure 1b. Moreover, initial conditions and boundary conditions are required for the computation. In the results section we will use data-sets that were computed using more sophisticated solvers and show that this is no crucial restriction. The third requirement ensures that exact conclusions about the influence of different positions and variables can be made.

Looking closer at these three requirements, we see that they correspond to the construction rules of cellular automata, which are well researched.

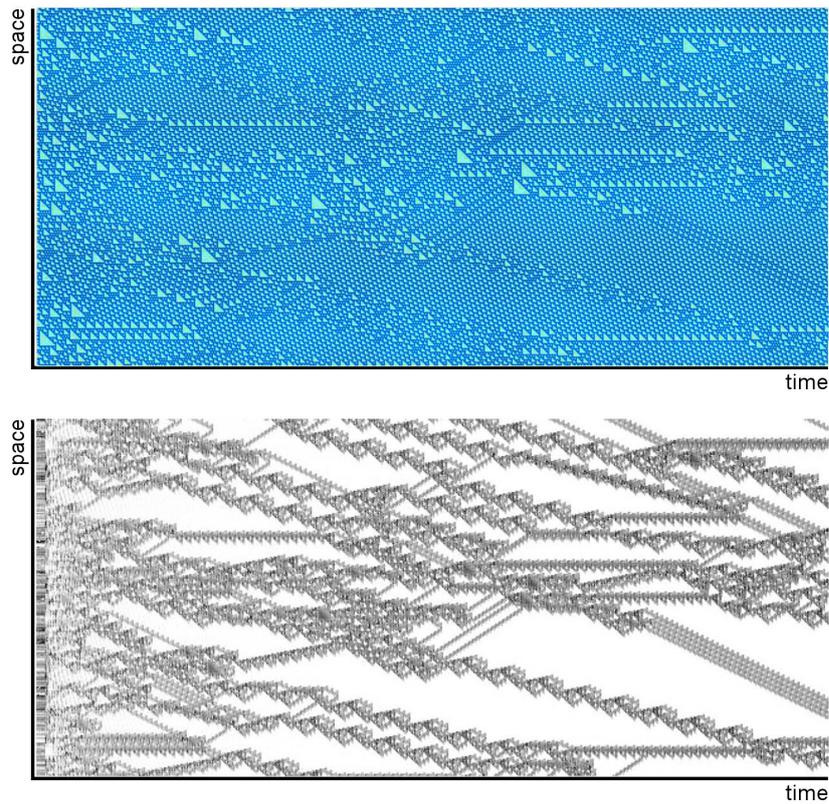


■ **Figure 1** Different structures in a Cartesian grid: (a) Empty grid. (b) Sample neighborhood used to compute finite differences. (c) Light-cone structures used for the computation of local statistical complexity.

4 Cellular Automata

A cellular automaton (CA) is a discrete model of a system, with the game of life being the best-known example. The automaton consists of a regular uniform lattice with a discrete variable at each cell. The configuration of an automaton at a certain time step is completely specified by the values of the variables at each site. Following predefined local rules the configuration can change at each discrete time step. A rule defines which value a cell will take in the next step, depending on the values of its neighborhood in the present. Typically, the neighborhood of a cell consists of the cell itself and all immediately adjacent cells. An example for a rule is: If the cell has value 0 and at least two of its neighbors have value 1, change the cell's value to 1. For each time step all values are updated simultaneously.

An example of a 1D cellular automaton is given in Figure 2. The domain can be separated into two different classes: stable local patterns and defects. The stable local patterns are the areas, that look like the background of the image. The defects are the triangles in different sizes that move across the image. Although the stable patterns dominate after some time, the defects are the ones that determine the long term behavior. Shalizi et al. [26] proposed a filter for the automatic extraction of coherent structures, i.e. defects, in CA. Their filter is called local statistical complexity and automatically



■ **Figure 2** Cellular automaton in 1D (top) and corresponding local statistical complexity field (bottom).

detects prominent formations of arbitrary size and shape in unsteady data-sets. Figure 2(bottom) shows the filtered image of the 1D cellular automaton, highlighting the defects that move around in the original data-set.

5 Local Statistical Complexity

Local statistical complexity extracts those regions in an unsteady field, where a lot of information from the local past is required to predict the dynamics in the local future. This happens where the temporal evolution is very unusual compared to what happens in the rest of the field. In general, users are interested in a subset of these distinctive regions, as they know the basic structure of their data-set and want to find regions that behave differently. Especially for large intricate and little understood data-sets local statistical complexity is a helpful tool to guide the user to regions that might be relevant for him or her.

Local statistical complexity focuses on the local temporal evolution of the field. The local past of position p in the field consists of all the points that might influence p . As effects propagate at finite speed, the past has the shape of a light-cone that is directed towards the past. The apex is located at p . This concept is likewise used when computing simulations using finite differences or finite elements. Here the value at position \vec{x} in time-step t is computed from the neighborhood of the point in the previous time-step $t - 1$ (Fig. 1c). (An exception is pressure in incompressible flow.) The future is given by a light-cone that is directed in the opposite direction, i.e., the future. Each light-cone comprises a set of positions. The values at these positions together with the neighborhood information are called a configuration. A configuration can be thought of as a pattern that extends in time, space

and if appropriate over multiple variables. By definition future configurations contain the value at the apex, past configurations do not.

For each past-cone configuration we would like to be able to predict, what might happen in the future. The only value that we can predict exactly, is the one at the future-cone apex, as it results from the calculation rule of the simulation method (remember Fig. 1b). To predict the remaining values in the future-cone, we need statistics. We group several similar past-configurations and compute a histogram over the different futures that occur. This estimated distribution tells us which future configurations are likely for this particular class of behavior in the past. This procedure is repeated for all different groups of past-configurations.

Analyzing the histograms we computed in the previous step, we will observe that some of them are very similar. This means that the differences we detected in the past-configurations have no significant influence on the dynamics in the future. Thus, we merge all those past groups that have very similar histograms. The different groups that result after the merging are called causal states. A causal state represents a cause-and-effect relationship between what was observed in the past and what might happen in the future. So, if we have a past configuration and can determine its causal state, we can estimate the most probable future dynamics.

Now that we can predict the dynamics in the future given the past configuration, we want to find a minimal lossless encoding for this information. The code with the shortest expected length is given by a Huffman-code. A Huffman-code assigns frequent symbols short codewords and rare symbols longer ones. The entropy $H[X]$ is a measure of the smallest average codeword length that is theoretically possible for the given alphabet X . For functions $f(x) x \in X$, mutual information $I[f(X), X]$ equals entropy $H[f(X)]$. In order to find an optimal encoding for the past-configurations, we have to find a function f that minimizes the mutual information $I[f(PastConf); PastConf]$. Shalizi et al. [26] showed that the unique function that minimizes the mutual information is the mapping to the causal states. Thus, if we store at each position the Huffman-code of the corresponding causal state, we resolve the file with shortest expected length that still gives us all informations about the dynamics in the local future.

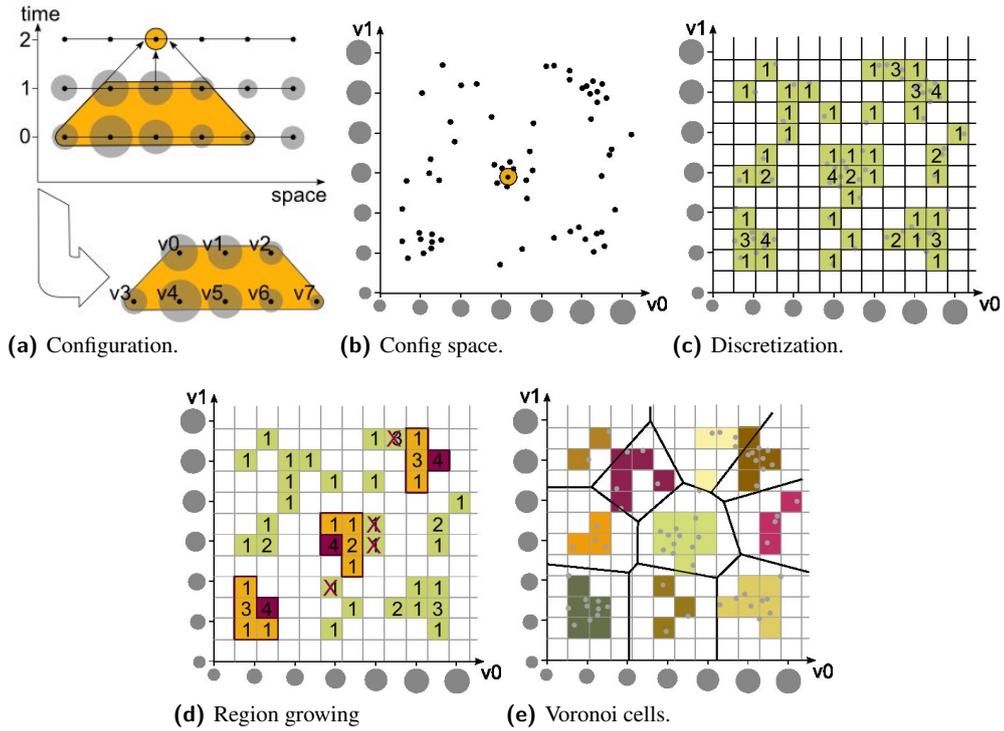
The encoded file can finally be used to detect distinctive regions. The Huffman-code assigns each causal state a codeword whose length depends on the number of positions that are assigned to it. Causal states with a very long codeword feature dynamics in the future that occur very rarely in the field. Local statistical complexity measures for a past-configuration the length of the codeword of the corresponding causal state, i.e., the amount of information that is needed to predict the causal state/the dynamics in the future. The longer the codeword, the more likely it is that something extraordinary is going to happen in the local future of this position. More information on the theory and implementation of local statistical complexity and causal states can be found in [26, 14].

6 Application to Finite Difference Schemes

Complexity analysis using local statistical complexity can be applied to scientific simulations as finite difference schemes, a direct analog to CA rules, can be used to discretize PDEs. The following simple example of an isotropic diffusion, e.g., ion concentration in water, is used for illustrations. Given a concentration $f(\vec{x}, t_0)$ at each position $\vec{x} \in B$ at time t_0 , the temporal development of this concentration $f(\vec{x}, t)$ is observed. The governing PDE is

$$\frac{\partial f}{\partial t}(\vec{x}, t) = D\Delta f(\vec{x}, t) \quad (1)$$

with a constant diffusion coefficient D , time derivative $\frac{\partial f}{\partial t}(\vec{x}, t)$ and Laplacian $\Delta f(\vec{x}, t)$. As boundary conditions constant concentrations are assumed: $f(\vec{x}, t) = f(\vec{x}, t_0)$ for $x \in \partial B$. A simple finite



■ **Figure 3** Density-driven Voronoi tessellation: (a) A past configuration extracted from the data-set consisting of eight variables. (b) This configuration marked in high-dimensional configuration space (only first two of the eight variables illustrated). (c) Initial fine-grained discretization of the configuration space. (d) Density-driven region growing starting in densest regions. (e) Final Voronoi tessellation of the configuration space. (f) Final partitioning of the domain.

difference scheme in the plane consists of a Cartesian lattice $L = \{0, \dots, 255\} \times \{0, \dots, 255\}$, a given concentration $f_0 : L \rightarrow \mathbb{R}$, and the difference equation

$$\begin{aligned}
 f(x_1, x_2, t+1) = & \frac{1}{16}f(x_1-1, x_2+1, t) + \frac{1}{8}f(x_1, x_2+1, t) + \frac{1}{16}f(x_1+1, x_2+1, t) + \\
 & \frac{1}{8}f(x_1-1, x_2+0, t) + \frac{1}{4}f(x_1, x_2+0, t) + \frac{1}{8}f(x_1+1, x_2+0, t) + \quad (2) \\
 & \frac{1}{16}f(x_1-1, x_2-1, t) + \frac{1}{8}f(x_1, x_2-1, t) + \frac{1}{16}f(x_1+1, x_2-1, t)
 \end{aligned}$$

which is also known as applying a binomial 3×3 filter to a digital image in image processing [12]. In this example L is the lattice of the CA, f contains the values over time and Eq. 2 gives the complete rule. As $c = 1$, the configurations are as illustrated in Fig. 1c. The reader familiar with either finite difference schemes or image processing might imagine a larger stencil or filter for $c > 1$. Similar schemes can be applied to any PDE, allowing for analysis using local statistical complexity.

7 Computation of Local Statistical Complexity

The first step in visualizing the local statistical complexity of a data-set consists of the computation of causal states. Causal states are defined by:

$$\text{Causal State} = \varepsilon(I^-) = \{\lambda : P(I^+|\lambda) = P(I^+|I^-)\}. \quad (3)$$

Hence, a causal state is the equivalence class of all past-cones (I^-) that have the same distribution ($P(I^+|I^-)$) over possible futures (I^+), i.e., each causal state predicts a certain future and the possible futures of different causal states differ.

To determine the causal states that occur within a data-set, the conditional probabilities $P(I^+|I^-)$ have to be estimated. As exactly the same pattern I^+ or I^- commonly only occurs once in a scientific data-set, the probabilities cannot be estimated directly, but similar configurations have to be grouped for the estimation. The grouping has to fulfill two requirements. First, all samples in the data-set have to be assigned to a group and second, the size of each group in high-dimensional space (dimensionality is given by the number of entries in the cone, cf. Fig. 1c) has to be the same to allow for a correct estimation.

In [13], Jänicke et al. proposed a fast strategy to estimate probabilities with a single sweep through the data. We use this approach based on density-driven Voronoi tessellation, which consists of three steps:

1. *Discretization*: Compute the past- and future-cone (Fig. 3a) at each position and store the discretized cones in two trees.
2. *Density-driven Voronoi Tessellation*: Partition the high-dimensional discrete cone space (Fig. 3b) using a Voronoi tessellation (Fig. 3(c-e)) that takes the underlying distribution of cone configurations into account. This step is performed for the past and future tree separately. Resulting IDs are stored for each leaf in the two trees.
3. *Probability Estimation*: For each past cell, the corresponding future cells are counted and used to estimate the probabilities.

The idea behind density-driven Voronoi Tessellation is to let the discretization adapt to the structure of the high-dimensional data. The initial discretization in Step 1. is used to estimate a local density. Starting from densest regions, a region growing algorithm is applied that iteratively captures the entire space. The method ensures that the Voronoi cells have equal size and that clusters are well preserved.

To identify causal states, the conditional probabilities have to be estimated. This is achieved by counting the number of occurrences of different future classes per past Voronoi cell. Dividing by the total number of configurations per past cell, gives the conditional probability $P(I^+|I^-)$. In a last step, those Voronoi past cells are grouped that feature a similar distribution over futures using a χ^2 -test [10]. The resulting grouped classes are the causal states of the process.

Each of these causal states represents a spatio-temporal pattern, indicating what might happen next if a certain past was observed. After the identification of the causal states, new fields that hold the ID of the causal state at each position are created. As we are not interested in the local pattern but in the complexity of the current position, we have to evaluate the local statistical complexity of each causal state and assign appropriate values to the field of causal state IDs.

Local statistical complexity measures how much information from the local past is required to predict the dynamics in the local future at a certain position. If the dynamics of a configuration match the average behavior in the data-set, only little information is required. On the contrary if something unusual happens, more information is required. To measure how extraordinary some local dynamics are, Shalizi et al. [26] proposed local statistical complexity, which was extended to scientific simulation data by Jänicke et al. ([14, 13]). The local statistical complexity at a certain position p in the field is defined as the mutual information between the corresponding configuration's past (I^-) and its causal state ($\mathcal{E}(I^-)$):

$$LSC(p) = I[\mathcal{E}(I^-); I^-]. \quad (4)$$

Mutual information is a measure from information theory, which tells how much information one

random variable contains about another one:

$$I[A;B] = \sum_{a \in A; b \in B} P(a,b) \log_2 \frac{P(a,b)}{P(a)P(b)} \quad (5)$$

where $P(a)$ is the probability that the random variable A takes the value a and $P(a,b)$ is the corresponding joint probability of variables A and B . Using this definition, the local statistical complexity of a cone configuration tells how much information from the past is required to identify its causal state. If one knows the causal state, the dynamics in the future are clear as well. Hence, if a lot of information is required to identify the causal state, the local dynamics are extraordinary compared to what is happening in the rest of the data-set.

8 Results

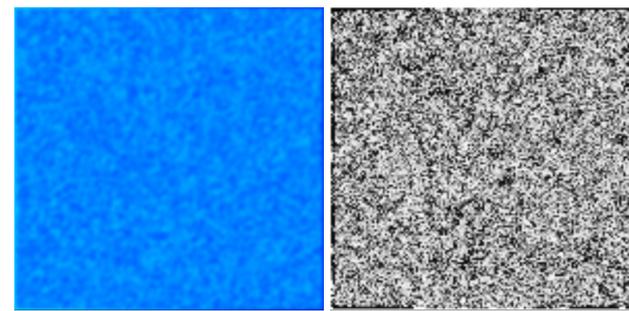
The three data-sets we are going to analyze have increasing complexity. The first one is an isotropic diffusion which is a perfect analogon to CA. In the second example we will analyze swirling flow. This 2d examples consists of multiple variables and contains different features experts are interested in. The third test-case is a simulation of the flow around a delta wing. In this large 3d example several intricate features are present. The results of local statistical complexity will be compared to standard feature detection techniques for both CFD examples to verify the correctness of the automatic detected features.

8.1 Isotropic Diffusion

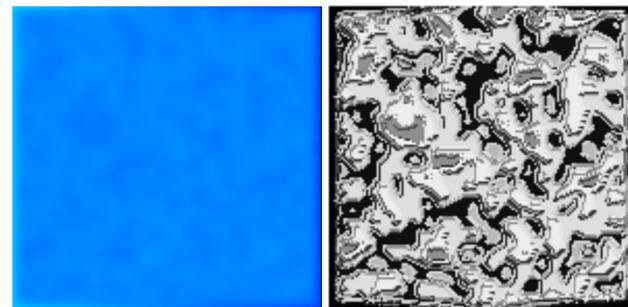
An isotropic diffusion, simulated using finite differences as explained in Section 6, is a simple example of a large variety of diffusion processes, i.e., equalization of differences in concentration, heat, matter or momentum, appearing in nature. The dataset is simulated by repeated filtering using a binomial filter. In the diffusion field, the cells at the left border are set to 1, and those at the right border to 0. Upper and lower boundaries are initialized with linearly decreasing values that range from 1 to 0. The inner part is initialized with random values between 0.0 and 1.0. The process displayed in the upper row of Fig. 4, is defined on a square lattice with 150 cells in each direction. 800 time-steps are simulated.

The left half of the images in Figure 4 shows the evolution of the diffusion. In time-step 1 the image consists of many small coherent structures that still feature a large variety of values. After 20 time-steps these homogeneous regions have become much larger and the range of values has shrunk. At the boundaries small bands with the extremal values are visible. This process continues in time-step 50. While the center becomes more homogeneous, the gradients starting from the boundaries grow. In time-step 800 half of the domain has reached the equilibrium of the diffusion process.

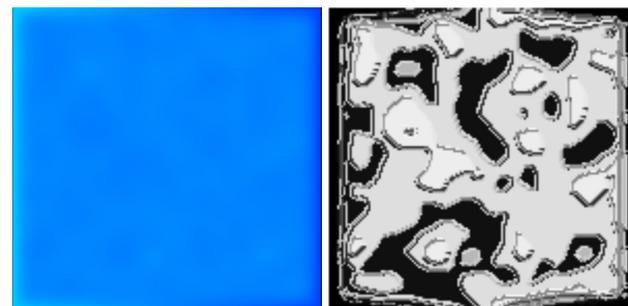
On the right hand-side of this series of snapshots, the corresponding complexity fields are depicted. In the first time-step the entire domain is covered by small black and gray spots. The areas that appear in light gray, are those that hold values close to 0.5, the most common value in this data-set. Black cells hold formations that have either very different or extremal values in their configurations. In time-step 20 the diffusion has formed larger homogeneous regions, which are found by local statistical complexity. Again, black areas indicate extreme values and gray areas normal ones. These areas grow (time-step 50) until in time-step 200 the entire center of the data-set holds value 0.5. Thus, this pattern is the basic one and considered to be uninteresting. In time-step 800 the gradient covers half of the data-set. As we only analyze those time-steps in which the gradient evolves, these configurations with increasing/decreasing values are something extraordinary, whereas configurations containing only value 0.5, the standard result of the diffusion process, are considered to be normal.



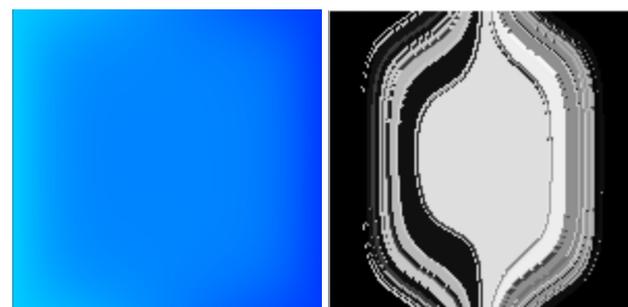
(a) Time-step 1.



(b) Time-step 20.

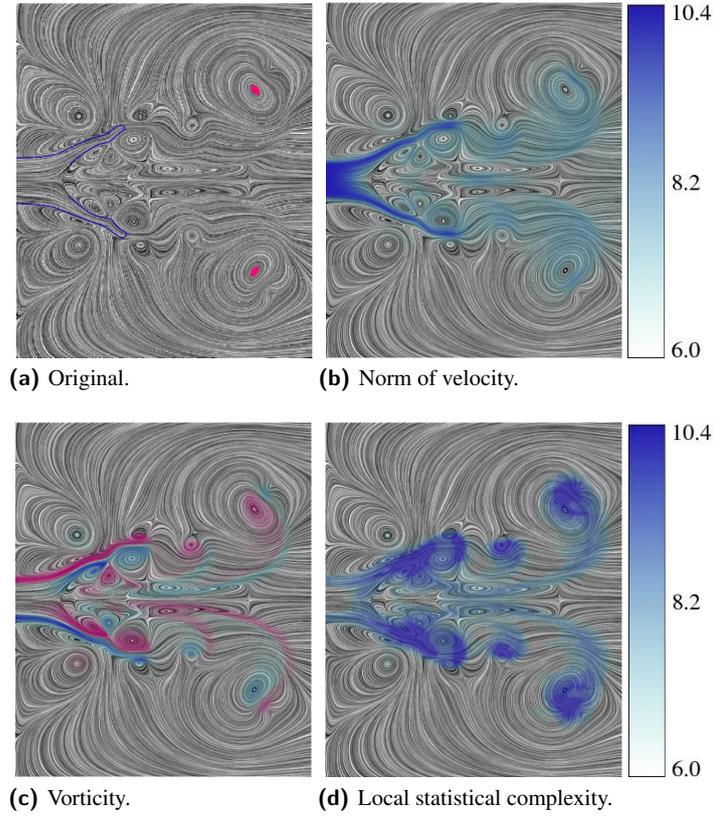


(c) Time-step 50.



(d) Time-step 800.

■ **Figure 4** Evolution of the diffusion data-set (left - original values, right - lsc field): (a) (left) Random initialization in time-step 1. (right) Local patterns that are close to equilibrium occur in light-gray in the lsc-field, which indicates small complexity. Pixels colored in black include extremal values which seldom occur in the entire unsteady data-set. (b) Time-step 20: Coherent structures start to form. (c) Time-step 50: Coherent structures in the center grow. Large areas reach equilibrium (light-gray in center). The gradient grows starting from the boundary. (d) Time-step 800: The center has reached equilibrium (value 0.5) and the boundary gradient grows further.



■ **Figure 5** Swirling flow: In each image the line integral convolution (LIC) of the velocity field is overlaid with an additional quantity. (a) The conical shear region (blue) outlines the region where the flow enters the domain. Two red points mark one of the ring-like vortex structures. (b) The norm of velocity overlay highlights regions with a strong current and reveals the relevant structures. (c) Vorticity indicates strong swirling motion. The color-coding gives the orientation. (d) Local statistical complexity automatically extracts analog structures.

8.2 Swirling Flow

The development of a recirculation zone in a swirling flow is investigated by numerical simulation. This type of flow is relevant to several applications where residence time is important to enable mixing and chemical reactions.

The unsteady flow in a swirling jet is simulated with an accurate finite-difference method. The Navier-Stokes equations for an incompressible, Newtonian fluid are set up in cylindrical coordinates assuming axi-symmetry in terms of streamfunction and azimuthal vorticity. All equations are dimensionless containing the Reynolds number Re and the swirl number S as defined by Billant et al. [3]

$$Re \equiv \frac{v_z(0, z_0)D}{\nu} \quad S \equiv \frac{2v_\theta(R/2, z_0)}{v_z(0, z_0)} \quad (6)$$

where $z_0 = 0.4D$, $D = 2R$ is the nozzle diameter and ν the kinematic viscosity, as dimensionless parameters.

The PDEs are discretized with fourth order central difference operators for the non-convective terms and with a fifth order, upwind-biased operator [17] for the convective terms. The time integrator is an explicit s -stage, state space Runge-Kutta method ([5], [15]), the present method is fourth order

accurate with $s = 5$. The time step is controlled by the minimum of two criteria: The limit set by linearized stability analysis and the limit set by the error norms of an embedded third order Runge-Kutta scheme [5]. The Helmholtz PDE for streamfunction $\tilde{\Psi}(r, z, t)$ is solved with an iterative method using deferred corrections and LU-decomposition of the coefficient matrix. The deferred corrections method is designed to reduce the bandwidth of the coefficient matrix. It converges rapidly using about ten to twenty steps.

The flow domain is the meridional plane $\mathcal{D} = \{(r, z) : 0 \leq r \leq R, 0 \leq z \leq L\}$ with $R = 5D$, $L = 8D$ and D denoting the nozzle diameter at the entrance boundary. The flow domain is mapped onto the unit rectangle which is discretized with constant spacing. The mapping is separable and allows to a limited extent crowding of grid points in regions of interest. The present simulation uses $n_r = 91$ and $n_z = 175$ grid points in radial and axial directions. The boundary conditions are of Dirichlet type at the entrance section and the outer boundary and at the exit convective conditions are imposed for the azimuthal vorticity. The initial conditions are stagnant flow and the entrance conditions are smoothly ramped up to their asymptotic values within four time units.

The simulation results for $Re = 10^3$, $S = 1.1$ (within the range of the experiments [3], [18]) used for the complexity analysis are ten time steps after the formation of the recirculation bubble (which forms at $t = 6.02$) at times $t = 33.63092$ to $t = 33.70560$. The flow is unsteady and does not approach a steady asymptotic state as the velocity and vorticity fields show (Fig. 5(a-c)).

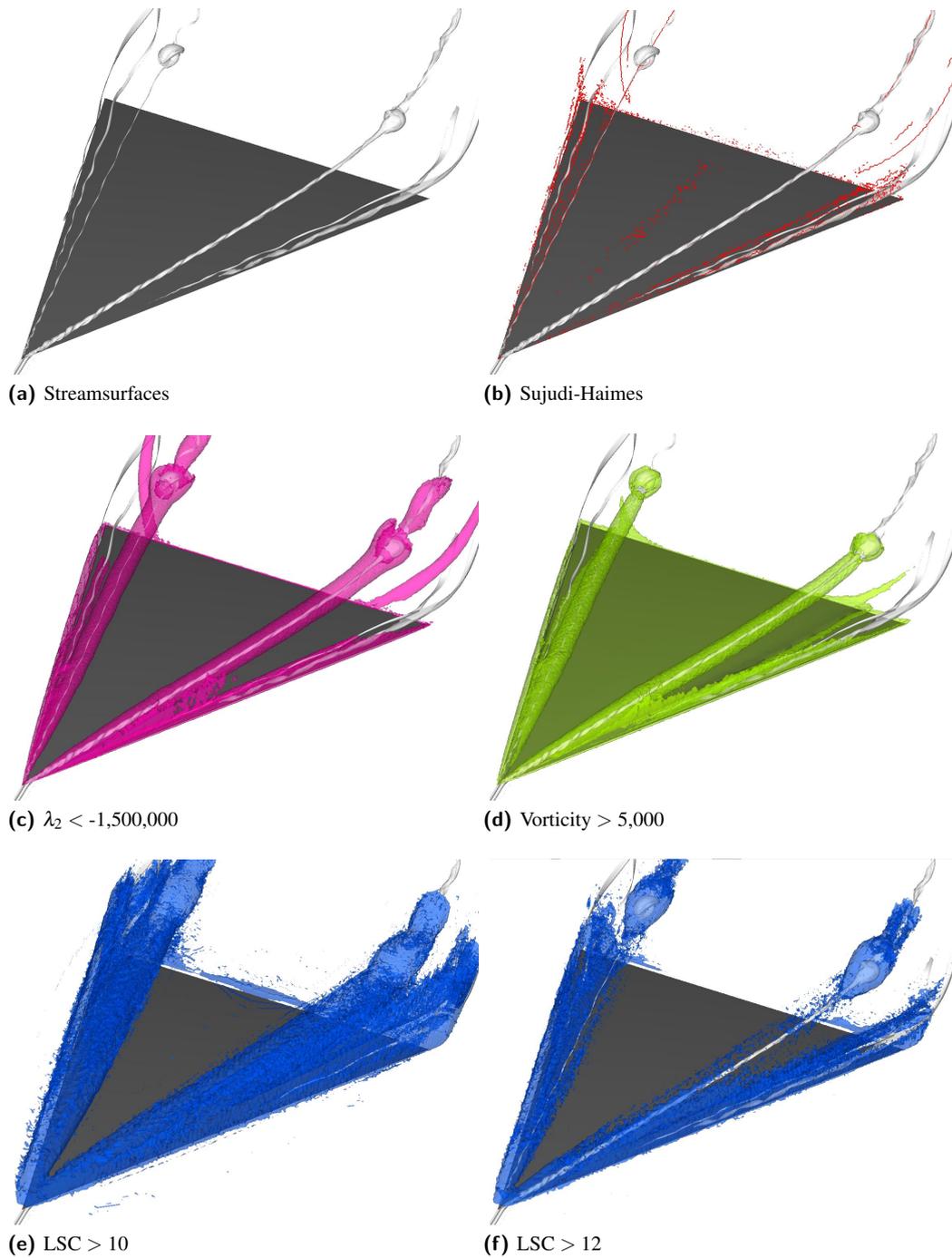
Figure 5(a) shows a line integral convolution (LIC) of the velocity field, featuring several vortices. Relevant features are highlighted in this image. The structure outlined in blue is the conical shear region surrounding the inlet of the swirling flow. The two red dots indicate one ringlike vortex structure. The coreline of this vortex lies in a plane orthogonal to displayed cross-section and passed through the red points. Comparing this image to the one overlaid with the norm of velocity (Fig. 5b), we see that the simple LIC image gives a misleading impression of the flow as several of the clearly visible vortices are detected in regions close to noise.

The vorticity overlay in Figure 5c results in a similar image as norm of velocity. Basically the same structures are highlighted. Differences occur at the inlet, where, as expected by the technique, only the shear flow is highlighted. Moreover, the ring-like vortex structures are more pronounced than the connecting structures. The color-coding provides an additional hint telling the orientation of the rotation.

Local statistical complexity is computed for the combination of velocity and vorticity and layed over the original LIC to provide context (Fig. 5d). Both features, the shear region and the ring-like vortex structures, are automatically detected by local statistical complexity. Unlike vorticity, local statistical complexity marks both features as equally complex. Both, the conical shear region, as well as the vortex structure are assigned highest complexity, while the vortices exhibit only small vorticity, compared to the shear flow.

8.3 Delta Wing

This data-set represents the airflow around a delta wing at low speeds with an increasing angle of attack. Multiple vortex structures form on the wing due to the rolling-up of the viscous shear layers that separate from the upper surface. These formations of three vortices can be observed on either side of the wing (Fig. 6a). With increasing angle of attack the intensity of the primary vortices (purple) increases until in time-step 700 a vortex breakdown occurs. This phenomenon is characterized by rapid deceleration of both the axial and tangential mean velocity components inside the vortex. During breakdown, the axial mean velocity component vanishes and becomes negative on the axis of the vortex, corresponding to appearance in the flow structure of a stagnation point followed by a recirculation bubble. The analysis of vortex breakdown is highly interesting, as it is one of the limiting factors of extreme flight maneuvers. The extraction and visualization of the individual



■ **Figure 6** Delta Wing: (a) Streamsurfaces to indicate the vortices above the delta-wing. (b) Sujudi-Haimes vortex detection algorithm applied to the vector field. (c) Isosurface in the λ_2 -field (isoval = -1,500,000). (d) Isosurface in the vorticity field (isoval = 5000). (e,f) Isosurface in the local statistical complexity field of the norm of velocity (Isoval = 10 (e), Isoval = 12 (f)).

structures, however, is still a challenging task as the different structures are nested and interact with each other. The unstructured grid was resampled on a $292 \times 224 \times 75$ grid (~ 4.1 Million positions) and consists of more than 1000 time-steps. The images in Figure 6 depict time-step 700.

Images 6(b-d) give an overview over standard vortex detection techniques. The algorithm by Sujudi and Haines [29] (Fig. 6b) is a technique that detects vortex core-lines. Applied to the delta wing, this method perfectly extracts the core-line of the major vortices. However, we only get a vague indication of the core-lines close to the surface, whose vortices are less dominant and interact with each other. The λ_2 -criterion extracts the “hull” of the vortex. Finding an appropriate isovalue ($-1,500,000$) to separate the two minor vortices without missing the recirculating bubble takes some time. The isosurface of the magnitude of the vorticity (Fig. 6d) gives approximately the same result.

Figures 6(d-f) show the local statistical complexity of the norm of velocity. Figure 6e shows all positions that are assigned a complexity value greater than 10 (maximum: 14.7). The visualized structures do not only comprise the vortices and the recirculation bubble, but also the regions at the outer corners of the wing, where the flow from the smaller vortices and the flow from underneath the wing interact and form a swirling motion that is classified by the other techniques as vortex. Increasing the complexity value further (Fig. 6f), we see that the individual vortices are better separated. The major vortices are no longer visible as their complexity value is smaller than those of the small vortices. This observation means, that the local temporal evolution of the norm of the velocity is very distinct for vortices and for the recirculating bubble. The exceptional behavior of the norm of the velocity is a typical characteristic for recirculating bubbles, as was explained earlier. With our method we can extract these distinctive formations automatically without defining a definite pattern beforehand. This feature is an important characteristic of our method, as it is capable of identifying structures that exhibit an extraordinary formation without precisely describing its pattern.

9 Conclusion

In this paper we described a filter called local statistical complexity based on concepts from information theory which automatically extracts coherent structures from unsteady multi-fields. It assigns each position in the data-set a scalar value whose magnitude depends on how extraordinary the local dynamics at the current position are. Color-mapping or isosurfacing can be used to visualize the most distinct structures in the data-set.

Local statistical complexity is intrinsic to unsteady multi-field visualization as this is required by the theory and quantities of different type (scalar, tensor, vector valued) can be used simultaneously in the computation. The process reduces the multi-field to a single scalar field giving the importance of each position. The entire process is fully automatic and requires no application-specific knowledge. (The user has to provide two parameters for the Voronoi tessellation, which could be estimated as well.)

Current problems arise, when analyzing divergence-free flow, as the concept of local influence propagation is not preserved. When the dynamics are too turbulent, memory costs increase a lot, as many different configurations have to be stored. The alternative is to compute coarser causal states, which makes the results more inaccurate.

In our future work we would like to work on a complete mathematical basis in the continuous case. More research has to be done regarding the influence of the parameters in the Voronoi tessellation process. The original concept was designed for PDEs solved using finite differences. Extending the theory to other numerical schemes is a further task that should be addressed in the future.

10 Acknowledgements

The authors wish to thank Wolfgang Kollmann for helpful discussion concerning the application of local statistical complexity to data-sets from fluid dynamics and for providing the swirling flow data-set. The delta wing data-set was provided by Markus Rütten from the German Aerospace Centre (DLR).

References

- 1 R Badii and A Politi. *Complexity: Hierarchical Structures and Scaling in Physics*. Cambridge University Press, Cambridge, 1997.
- 2 W. Bialek, I. Nemenman, and N. Tishby. Predictability, complexity, and learning. *Neural Computation*, 13:2409–2463, 2001.
- 3 P. Billant, J.M. Chomaz, and P. Huerre. Experimental study of vortex breakdown in swirling jets. *JFM*, 376:183–219, 1999.
- 4 James P. Crutchfield and Karl Young. Inferring statistical complexity. *Phys. Rev. Lett.*, 63(2):105–108, July 1989.
- 5 G. Wanner E. Hairer. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, 2nd ed.* Springer Verlag, Berlin, 1996.
- 6 Julia Ebling and Gerik Scheuermann. Clifford convolution and pattern matching on vector fields. In *Proceedings of the 14th IEEE Visualization 2003*, pages 193–200, Washington, DC, USA, 2003. IEEE Computer Society.
- 7 Christoph Garth, Florian Gerhardt, Xavier Tricoche, and Hans Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1464–1471, 2007.
- 8 Peter Grassberger. Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics*, 25(9):907–938, September 1986.
- 9 Einar Heiberg, Tino Ebbers, Lars Wigstrom, and Matts Karlsson. Three-dimensional flow characterization using vector pattern matching. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):313–319, 2003.
- 10 P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Statistical Theory: Ch. 3 - Testing Hypotheses*. New York: Houghton Mifflin, 1971.
- 11 T Hogg and BA Huberman. Order, complexity and disorder. *Xerox Palo Alto Research Center (preprint)*, 1985.
- 12 Bernd Jähne. *Digital image processing: concepts, algorithms and scientific applications*. Springer-Verlag, London, UK, 1991.
- 13 Heike Jänicke, Michael Böttlinger, Xavier Tricoche, and Gerik Scheuermann. Automatic Detection and Visualization of Distinctive Structures in 3D Unsteady Multi-fields. *Computer Graphics Forum*, 27(3):767–774, 2008.
- 14 Heike Jänicke, Alexander Wiebel, Gerik Scheuermann, and Wolfgang Kollmann. Multifield visualization using local statistical complexity. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1384–1391, 2007.
- 15 Christopher A. Kennedy, Mark H. Carpenter, and R. Michael Lewis. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. *Appl. Numer. Math.*, 35(3):177–219, 2000.
- 16 A. Kolmogorov. Logical basis for information theory and probability theory. *IEEE Trans. on Information Theory*, 14(5):662–664, September 1968.
- 17 Yuguo Li. Wavenumber-extended high-order upwind-biased finite-difference schemes for convective scalar transport. *J. Comput. Phys.*, 133(2):235–255, 1997.

- 18 H. Liang and T. Maxworthy. An experimental investigation of swirling jets. *Journal of Fluid Mechanics*, 525:115–159, February 2005.
- 19 Frits H. Post, Benjamin Vrolijk, Helwig Hauser, Robert S. Laramee, and Helmut Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.
- 20 M. Roth. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD Thesis, Swiss Federal Institute of Technology, ETH Zürich, 2000.
- 21 T. Salzbrunn, H. Jänicke, T. Wischgoll, and G. Scheuermann. The State of the Art in Flow Visualization: Structure Based Techniques. In *Proc. Simulation and Visualization (SimVis)*, pages 75–92, 2007.
- 22 T. Salzbrunn and G. Scheuermann. Streamline Predicates. *IEEE TVCG*, 12(6):1601–1612, 2006.
- 23 Geric Scheuermann and Xavier Tricoche. Topological Methods for Flow Visualization. In Charles. D. Hansen and Christopher R. Johnson, editors, *The Visualization Handbook*, pages 341–358. Elsevier, Amsterdam, 2005.
- 24 Michael Schlemmer, Manuel Heringer, Florian Morr, Ingrid Hotz, Martin Hering-Bertram, Christoph Garth, Wolfgang Kollmann, Bernd Hamann, and Hans Hagen. Moment invariants for the analysis of 2d flow fields. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1743–1750, 2007.
- 25 Cosma Rohilla Shalizi. Optimal nonlinear prediction of random fields on networks. In Michel Morvan and Éric Rémila, editors, *Discrete Models for Complex Systems, DMCS'03*, volume AB of *DMTCS Proceedings*, pages 11–30. Discrete Mathematics and Theoretical Computer Science, 2003.
- 26 Cosma Rohilla Shalizi, Robert Haslinger, Jean-Baptiste Rouquier, Kristina Lisa Klinkner, and Cristopher Moore. Automatic filters for the detection of coherent structure in spatiotemporal systems. *Physical Review E*, 73:036104, 2006.
- 27 C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, July 1948.
- 28 K. Shi, H. Theisel, H. Hauser, T. Weinkauff, K. Matkovic, H.-C. Hege, and H.-P. Seidel. Path Line Attributes - an Information Visualization Approach to Analyzing the Dynamic Behavior of 3D Time-Dependent Flow Fields. In *Topology-Based Methods in Visualization, Proceedings of the 2007 Workshop (to appear)*, 2007.
- 29 D. Sujudi and R. Haimes. Identification of swirling flow in 3d vector fields. *AIAA 95-1715*, 1995.