Multifield Visualization Using Local Statistical Complexity

Heike Jänicke, Alexander Wiebel, Gerik Scheuermann, *Member, IEEE CS*, and Wolfgang Kollmann

Abstract— Modern unsteady (multi-)field visualizations require an effective reduction of the data to be displayed. From a huge amount of information the most informative parts have to be extracted. Instead of the fuzzy application dependent notion of feature, a new approach based on information theoretic concepts is introduced in this paper to detect important regions. This is accomplished by extending the concept of local statistical complexity from finite state cellular automata to discretized (multi-)fields. Thus, informative parts of the data can be highlighted in an application-independent, purely mathematical sense. The new measure can be applied to unsteady multifields on regular grids in any application domain. The ability to detect and visualize important parts is demonstrated using diffusion, flow, and weather simulations.

Index Terms—Local statistical complexity, multifield visualization, time-dependent, coherent structures, feature detection, information theory, flow visualization.

1 Introduction

One of the strengths of scientific visualization is the fact that it can communicate large amounts of information. However, images can become too crowded and cluttered to see the important facts, too. Highlighting the most informative regions is a powerful method to reduce the amount of information displayed and guide the observers attention.

The crucial step is the extraction of these regions, often called features. At present, there exists no general definition of a feature, except for being a structure or region of relevance. Depending on the application, e.g., computational fluid dynamics (CFD), electromagnetic field simulations, weather models or simulations of biological systems, completely different structures are of interest. In general, features are detected by searching regions that fulfill certain criteria, e.g., exhibit a certain value or pattern [29]. As these criteria have to be specified in advance, further problems arise. In the case of vortex detection, for example, no general definition exists. Vortices, however, are not of interest because they exhibit some kind of swirling structure, but as they have great impact on the behavior of the flow. This property is true for all features - they are of special importance to the system. Moreover, inside these structures it is hard to predict the system's behavior. Starting from this point, a feature can be characterized as a region that requires a lot of information about the past to predict its own dynamics. Utilizing information theory, an objective and universal definition of a feature can be given.

Shalizi et al. [36] proposed a local criterion, called local statistical complexity, to measure complexity in cellular automata (CA). Therewith features are identified with regions of high local statistical complexity. As stated in [28, 39], CA can be used to model partial differential equations (PDEs), and vice versa. The combination of these two approaches results in a general feature extraction method applicable to any kind of system that can be described by PDEs, which comprise the majority of computational science and engineering simulations. Besides being a generally applicable measure, two additional advantages lie in the nature of statistical complexity that many other feature detection methods do not have: Firstly, the method is based on time-dependent analysis, and therefore, can easily cope with unsteady datasets. And secondly, from an information theoretic point of view multifield analysis is required and directly realized if the system is influenced by several independent variables. In either case, the result of the analysis is a single time-dependent scalar field, defining importance in the dataset.

- *Heike Jänicke, Alexander Wiebel, and Gerik Scheuermann are with the Image and Signal Processing Group, Department of Computer Science, University of Leipzig, Germany. E-mail: {jaenicke, wiebel, scheuermann}@informatik.uni-leipzig.de.*
- *Wolfgang Kollmann is with the Department of Mechanical and Aeronautical Engineering, University of California, Davis. E-mail: kollmann@evolene.engr.ucdavis.edu.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 2 November 2007.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

2 Related Work

Most datasets obtained by measurements or numerical simulations are combinations of different quantities. Though many methods in the field of scientific visualization focus on the analysis of a single field, there are several approaches treating multiple fields at a time. In general, two different approaches can be distinguished. Either several fields are visualized in combination, or relations between the different fields are displayed. Examples of the first approach are mostly combinations of different techniques, e.g., color coding, glyphs and/or partially transparent maps as used by Bair et al. [3] or Kirby et al. [22] in 2D, and modified volume rendering in 3D (Andreassen [1], Riley [30]). The analysis of correlation between different fields belongs to the second category and was researched amongst others by Kniss et al. [23] and Sauber et al. [33]. A summary of different techniques can be found in [40].

Many multifield methods suffer from cluttering if the number of fields gets too large. An efficient way to reduce cluttering is to concentrate on relevant structures or regions, so called features. Highlighting important regions does not only reduce the amount of data to be displayed significantly without losing important information, but also focuses the observers attention. Consequently, features have to be extracted first. As there is no general feature definition so far, there exists no universal method to detect them. In the literature three different approaches can be distinguished: image processing (e.g. Ebling et al. [9], Heiberg et al. [17]), topological analysis (e.g. Scheuermann et al. [34]) and physical characteristics (e.g. Garth et al. [11], Roth [32]). A detailed description for the field of flow visualization can be found in [29].

As long as structures are relatively simple, it is possible to define features. But as soon as the systems become more complex, such as anthills, human brains or chemical reactions, the definition of what is relevant gets more difficult. A first step is to measure the system's complexity. A large variety of measures are available fulfilling this task, e.g., [5, 7, 12, 24, 37]. Common measures originating from the analysis of strings of data are Shannon entropy [37] and algorithmic information [2]. 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 [19] state that complexity is small for completely ordered and completely disordered patterns and reaches a maximum inbetween. A

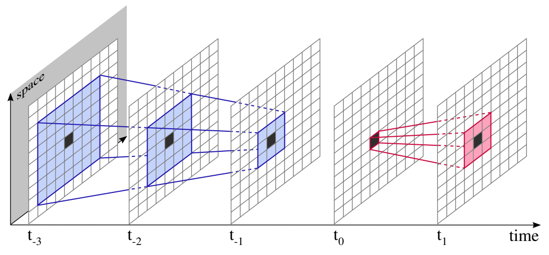


Fig. 1. Illustration of the past (blue) and future light-cone (red) of a single position (black cell). Influences propagate at speed $c = 1$, i.e., only immediately adjacent neighbors affect a cell. The cones are restricted to past-depth 3 and future-depth 2.

different approach was taken by Grassberger [12], who defined complexity as the minimal information that would have to be stored for optimal predictions. Based on this idea, statistical complexity [7] 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 analyse random fields, a point-by-point version was formulated by Shalizi [35] called local statistical complexity.

3 Local Statistical Complexity of Finite State CA

3.1 Cellular Automata

Local statistical complexity was introduced to detect coherent structures in cellular automata (CA). A 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.

CA were studied since the early 1950s and used to describe a large variety of systems, e.g., [13, 15, 16]. These models are commonly used to gain deeper insight into the underlying system, whereof many exhibit characteristic formations also known as coherent structures. Coherent structures are a result of complex patterns of interaction between simple units [4] and can be observed in CA as well. Identifying such structures automatically is still a challenging task. While most methods rely on previous knowledge about the strucures and search for regions fulfilling certain criteria, local statistical complexity identifies informative regions based on information theory.

3.2 Local Statistical Complexity

The basic idea of local statistical complexity is to identify spatiotemporal structures that exhibit the same behavior in the future, and measure their probability of appearance. The less likely a structure appears, the higher is its complexity. In the following, local statistical complexity as introduced by Shalizi et al. [36] is explained.

Let $f(\vec{x},t)$ be a discretized time-dependent n-dimensional field on a regular uniform lattice. In this field interactions between different space-time points propagate at speed c. Thus, all points possibly having influence on a point $p = (\vec{x},t)$ at time t are arranged in a so called light-cone as illustrated in Fig. 1. The same holds for all points being influenced by p. The past light-cone of p comprises all points $q = (\vec{y},\tau)$ with $\tau < t$ and $\|\vec{x}-\vec{y}\| \le c(t-\tau)$. The definition of the future light-cone is analogue. The configuration of the field in the past cone is denoted by $l^-(\vec{x},t)$, the one in the future cone by $l^+(\vec{x},t)$. Given a certain configuration l^-, different associated configurations l^+ might appear in the field, each with a certain probability. These probabilities are summarized in the conditional distribution $P\left(l^+|l^-\right)$. l^- contains all the information provided by the points in the past, which is often more than needed to predict l^+. Thus, l^- can be compressed using a function η, which defines a local statistic. The goal is to find a *minimal sufficient statistic* [25], i.e., a function with highest compression, that still allows for optimal prediction. How informative different statistics are, can be quantified using information theory. The *information* about variable a in variable b is

$$I[a;b] \equiv E\left[\log_2 \frac{P(a,b)}{P(a)P(b)}\right] = E\left[\log_2 \frac{P(a|b)}{P(a)}\right] \tag{1}$$

where $P(a,b)$ is joint probability, $P(a)$ is marginal probability, and $E[x]$ is expectation [25]. The information contained in $\eta\left(l^-\right)$ about the future is $I\left[l^+;\eta\left(l^-\right)\right]$. The minimal sufficient statistic ε is the function which maps past configurations to their equivalence classes, i.e., classes having the same conditional distribution $P\left(l^+|l^-\right)$:

$$\varepsilon\left(l^-\right) = \left\{\lambda | P\left(l^+|\lambda\right) = P\left(l^+|l^-\right)\right\} \tag{2}$$

As $P\left(l^+|\varepsilon\left(l^-\right)\right) = P\left(l^+|l^-\right)$, $I\left[l^+;\varepsilon\left(l^-\right)\right] = I\left[l^+;l^-\right]$, making ε a sufficient statistic. Minimality and uniqueness are proven in [36]. The equivalence classes $\varepsilon\left(l^-\right)$ are the *causal states* [7, 35] of the system, predicting the same possible futures with the same possibilities. As ε is minimal, the causal states contain the minimal amount of information needed to predict the system's dynamics. The minimal amount of information of a past light-cone needed to determine its causal state, $I\left[\varepsilon\left(l^-\right);l^-\right]$, and therewith its future dynamics, is a characteristic of the system, and not of any particular model. Accordingly, *local statistical complexity* is defined as

$$C(\vec{x},t) \equiv I\left[\varepsilon\left(l^-(\vec{x},t)\right);l^-(\vec{x},t)\right] \tag{3}$$

The local complexity field $C(\vec{x},t)$ is given by $-\log_2 P(s(\vec{x},t))$, with $s(\vec{x},t)$ being the causal state of space-time point at position $\vec{x}$ and time t. $C = 0$ holds if the field is either random or constant, and grows as the field's dynamics become more flexible and intricate. The complexity field can be computed in four steps:

1. Determine the past and future light-cones, l^- and l^+, up to a predefined depth.
2. Estimate the conditional distributions $P\left(l^+|l^-\right)$.
3. Cluster past light-cones with a similar distribution over future light-cones using a fixed-size χ^2-test ($\alpha = 0.05$) [18].
4. Calculate $C(\vec{x},t)$.

A more detailed description of the algorithm can be found in [36].

4 Local Statistical Complexity of Finite Difference Schemes

4.1 Example

Complexity analysis using local statistical complexity can be applied to scientific simulations as finite difference schemes, a direct analogue 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

$$\frac{\partial f}{\partial t}(\vec{x},t) = D\Delta f(\vec{x},t) \tag{4}$$

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

$$\begin{aligned} f(x_1,x_2,t+1) = {} & \\ & \frac{1}{16}f(x_1-1,x_2+1,t) + \frac{1}{8}f(x_1,x_2+1,t) + \frac{1}{16}f(x_1+1,x_2+1,t) + \\ & \frac{1}{8}f(x_1-1,x_2+0,t) + \frac{1}{4}f(x_1,x_2+0,t) + \frac{1}{8}f(x_1+1,x_2+0,t) + \\ & \frac{1}{16}f(x_1-1,x_2-1,t) + \frac{1}{8}f(x_1,x_2-1,t) + \frac{1}{16}f(x_1+1,x_2-1,t) \end{aligned} \tag{5}$$

which is also known as applying a binomial 3×3 filter to a digital image in image processing [20]. In this example L is the lattice of the CA, f contains the values over time and Eq. 5 gives the complete rule. As $c = 1$, the configurations are as illustrated in Fig. 1. 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.

4.2 Adaptations

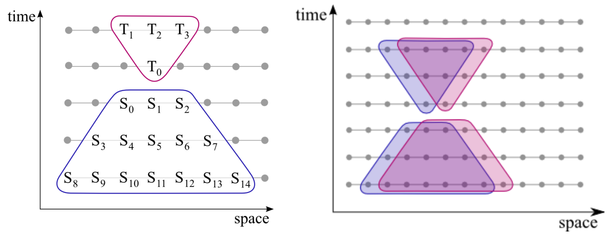
Local statistical complexity was designed to detect coherent structures in CA. Commonly, the cells of a CA can take only a few discrete values, e.g., from the set $\{0;1;2;3\}$. Thus, identical light-cones can be detected easily in Step 1 of the algorithm, as only a moderate number of configurations exist. If the approach is to be extended to time-dependent fields on regular grids generated by numerical simulations, this strict similarity has to be altered, as two configurations based on floating-point numbers hardly ever match exactly. Thus, light-cones with values within a certain range have to be considered equal, which necessitates a similarity measure for light-cones.

A first approach might be to simply discretize the fields. This method establishes strict arbitrary borders within the range of values, creating an artifical discrimination with no basis on the real data. Additionaly, only a few bits of information can be used to encode a floating-point number to have a realistic chance of observing identical cones. Thus, the discretization is very coarse, and moreover, destroys spatial and temporal correlation between the values, which is clearly visible in the complexity field. The same holds for clustering schemes employing kMeans or principle component analysis. Both methods (discretization and clustering) were tested, resulting in a complexity field that represents the discretization, and not the informative regions.

To avoid these problems a hierarchical method is used, subdividing the set of light-cones into different similarity classes iteratively. The optimized version of the algorithm is summarized in Algorithm 1. The extraction of past and future classifications is accomplished in two separate steps, each using the algorithm explained below. As structurally identical cones are required, only those points are considered that have light-cones completely inside the field, i.e., the first and last time steps are omitted according to the depth of the past and future light-cones respectively, and in each time step a boundary region of width max(pastDepth, futureDepth - 1)$\cdot c$ cannot be analysed.

From the points in the inner part of the field those are chosen as representatives, which exhibit largest differences in their configurations. Each representative stands for a similarity class, which comprises all configurations that are more similar to the current representative than to any other one. The resulting partitioning of the configurations corresponds to a Voronoi Diagram. This partitioning is constructed iteratively. The first representative is chosen randomly from all configurations in the restricted field. Afterwards, the distances between the first representative and all configurations are computed and stored. The configuration that is least similar to the first representative is chosen as second representative. Now, the representative/light-cone distances have to be updated. Therefore, all configurations that are more similar to the second representative than to the first are assigned a new, shorter distance. Thereafter, the third representative can be determined. This procedure is continued until a predefined number of representatives or a minimal distance between representatives and remaining light-cones is reached. The distance between each light-cone configuration and the closest representative is stored in a seperate vector, the so called shortest distance vector (SDV). Each entry of the vector belongs to a certain position in the field and holds the ID of the closest representative and the associated distance. The SDV is initialized by computing the distances between the first representative and all other configurations. In the update process entries are modified if the corresponding configuration is closer to the lastly added representative than to the stored one. The classification IDs needed for the estimation of conditional probabilities are stored in the SDV.

In order to compute distances between two configurations a distance measure and the way multifields are handled have to be defined. Let


time
space
T_1 T_2 T_3
T_0
S_0 S_1 S_2
S_3 S_4 S_5 S_6 S_7
S_8 S_9 S_{10} S_{11} S_{12} S_{13} S_{14}
time
space

(a) Sample cone configuration (b) Iterated light-cones

Fig. 2. **(a)** Illustration of two light-cone configurations in a 1D field. The red boundary encloses the future cone of T_0, the blue one the past cone of T_0. The elements of each cone are numbered in successive order. **(b)** Illustration of an iterated cone. The red cones are the spatial successors of the blue ones. The values in the overlapping regions can be reused for the second cone and need not be loaded from the field.

S_i^0 and S_i^1 be the configurations of light-cones C^0 and C^1 respectively, as illustrated in Fig. 2(a) for 1D. The distance between C^0 and C^1 is given by the L_2-norm:

$$\|C^0 - C^1\| = \sqrt{\sum_i \|S_i^0 - S_i^1\|^2} \tag{6}$$

$\|S_i^0 - S_i^1\|$ is the L_2-norm for scalar values and vectors. An elliptic similarity function to compare vectors was proposed in [38]. Unlike the L_2-norm, this measure favors certain directions, which foils the information theoretic approach. Moreover, the triangle inequality does not hold, whereby an important part of the optimization (see Sec. 5) is not applicable. Thus, the L_2-norm is taken as similarity measure.

If multiple fields are used, the cones are extended to multi light-cones MC, i.e., each multi light-cone consists of a vector of light-cones C_j, one for each field. To make the different subcones comparable, the fields have to be normalized in advance, i.e., each entry in the field has to be divided by the norm of the largest one. The joint distance is given by

$$\|MC^0 - MC^1\| = \sqrt{\sum_j \|C_j^0 - C_j^1\|^2} = \sqrt{\sum_j \sum_i \|S_{ij}^0 - S_{ij}^1\|^2} \tag{7}$$

Steps 2 to 4 of the algorithm can be applied directly, as they operate on the different classes of cones, and not on the individual values.

5 Efficient Implementation

The naive implementation of the light-cone classification is only applicable to fields of rather small size as demonstrated by the timings in Table 1. To handle larger time-dependent datasets in reasonable time, several aspects of an efficient implementation are proposed.

Whenever a cone is added to the list of representatives, all entries in the SDV have to be checked. Therefore, the distance between each cone in the field and the new representative is computed. It would be convenient to store all cone configurations in the field, which is not possible, since each cone consists of

$$\text{past: } \sum_{i=0}^{n-1} (3+2i)^d \quad \text{future: } \sum_{i=0}^{n-1} (1+2i)^d \tag{8}$$

elements, n being the depth of the cone, d the dimension of the lattice (2 or 3), and $c = 1$. Using light-cones of depth 3, would already multiply the memory usage by a factor of 83 in 2D, a crucial factor for large datasets with many time steps. As the configurations cannot be stored, they have to be rebuilt in each iteration. Using the fact that two succeeding cones have a large overlap, as illustrated in Fig. 2(b) for a 1D field, a single cone can be used that is iterated through the field. Thus, most of the time only a subset of the values has to be exchanged.

After the setup of the cone configuration, the distance to the new representative has to be computed, requiring 83 scalar or vector distances to be computed for cone depth 3, $d = 2$. As this has to be done for each representative and each position in each time step, the number of calculations of cone distances gets enormous. This can be reduced significantly by applying the triangle inequality.

$$\|r_c - r_n\| \leq \|l - r_c\| + \|l - r_n\| \tag{9}$$

Algorithm 1 Classification of light-cones

```
The field to be analysed is defined on a regular grid of
nbXPos×nbYPos positions, and nbTimeSteps time steps.
pastDepth ← depth of past light-cones
futureDepth ← depth of future light-cones
offset ← max( pastDepth, futureDepth - 1 )· c
n ← ( nbXPos - 2·offset ) · ( nbYPos - 2·offset ) ·
      ( nbTimeSteps - pastDepth - ( futureDepth - 1 ))

r ← randomly chosen representative
add r to the list of representatives
for i = 1 ... n do
   SDV[i] ← distance between r and the ith light-cone
   representative[i] ← 0
end for
init the list of candidates

while stopCriterion not fulfilled do
   if SDV[lastCandidate] > maxExcludedDist then
      add lastCandidate to the list of representatives
      update the distances of the candidates
   else
      update the SDV
      compute a new list of candidates
   end if
end while
update the SDV
```

The inequality compares the three distances between the following cone configurations: the current light-cone (l), the closest representative (r_c), and the new representative (r_n): Thus, the new representative can only be closer to the configuration than the old one ($\|l-r_n\| < \|l-r_c\|$) if Eq. 10 holds.

$$\|r_c - r_n\| < 2\|l - r_c\| \tag{10}$$

To employ this property, a matrix of all inter-representative distances is stored. Entry (i, j) of the matrix stores the distance between representatives i and j. Whenever a new representative is selected, an additional row and column has to be added to the matrix. Thus, both quantities of Eq. 10 are computed only once. $\|r_c - r_n\|$ is store in the matrix and $\|l - r_c\|$ in the SDV. The costly computation of the new distance $\|l - r_n\|$ is only performed if Eq. 10 holds. Thus, many comparisons can be omitted at very low cost.

So far, new representatives are chosen from all configurations in the field, although there are many configurations that are very similar to earlier defined representatives. Note that distances in the SDV can only become shorter by adding new representatives. If new representatives were chosen only from the configurations most dissimilar to selected representatives, the costs for the update can be further reduced. Accordingly, a list of potential representatives is stored. From the SDV the n cones with the longest distances are identified and their configurations and minimal distances are stored in a seperate list. n is a value specified by the user, e.g., if 3000 representatives are to be detected, lists of 600 cones were used. As an optimal choice of n depends highly on the dynamics of the field, no optimal value can be predefined. Additionally, the worst excluded distance is stored, which is the $n+1$st longest distance in the SDV. The iteration now only operates on the list of potential representatives. Whenever a new representative is added, only the minimal distances of the candidates are updated, which reduces the number of updates in a dataset with 100,000 positions from 100,000 to 600. Afterwards, a new representative is chosen from the list. This procedure is continued until the worst distance in the candidate list is shorter than the worst excluded one. As distances can only become shorter by adding new candidates, the worst excluded distance cannot become larger. For the given example of list length 600 approximately 150 representatives were added, before a new list had to be computed.

All improvements explained so far do not change the results, as they are just more efficient techniques and no heuristics. Nevertheless, for long time series the computational effort might still be too large. To decrease workload, the cone classification can be reduced to a subset of time-steps, i.e., representatives are chosen from every ith time-step. If the configurations in the field change at moderate speed, no essential structures are missed. The intermediate time-steps are classified in a second step. Again, the triangle inequality can be used to identify the closest representative. The shortest representative from the previous time-step is taken as an estimation of the shortest distance. From the list of representatives only those are compared to the current light-cone, that fulfill Eq. 10.

6 Results and Discussion

6.1 Influence of the Parameters

The computation of the causal states depends on two parameters: the depth of the light cones, and the number of representatives. The "Flow around a cylinder" (Section 6.5) is used to illustrate their influence.

Fig. 8 shows the complexity fields for different light-cone depths. The complexity field for the minimal configurations with past depth 1 and future depth 2 is shown in the third image. It already captures all relevant structures. Increasing the past depth to 2 results in smoother structures, as the region of influence becomes larger and single deviations are evened out. The difference between the two fields is shown in the fifth image. Blue regions indicate a higher complexity in the image with the smaller cones, red regions higher complexity in the image with deeper cones. The difference field is quite homogeneous, revealing the uniform modification of the field. Cones of depth 6 are used in the second image, which appears smoother than the other two. The difference field gives the same results as in the previous case. Thus, the depth of the cones has no significant influence on the result. The deeper the cones the smoother the image. To decrease computational costs, cones of depth 2 were chosen for all examples.

Just as the depth of light-cones, the number of representatives has no significant influence on the results. Fig. 6 shows the same dataset with different numbers of representatives used in the computation of complexity. If too few representatives are computed the relevant structures are visible but poorly developed. In the case of 8000 representatives certain regions are overrepresented, resulting in a kind of overfitting. Visually best results could be achieved when limiting the number of representatives to approximately 5000. Thus, the choice of the parameters has no great influence on the qualitative results, they mainly determine the quality of the resulting images.

6.2 Isotropic Diffusion

An isotropic diffusion, simulated using finite differences as explained in Section 4, 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. 3, is defined on a square lattice with 256 cells in each direction. 1202 steps are simulated.

The lower row in Fig. 3 shows the evolution of local statistical complexity. For each image 5000 representatives are determined, having past and future depth 2. The first image (top) shows the diffusion process at a very early stage, where large differences in concentrations are visible. Although, the image seems rather random, several dominating regions are already present as can be seen in the complexity field below. In the first few steps the concentrations of neighboring cells are equalized very quickly, forming a relative homogeneous region in the inner part (light blue). After 200 iterations, the boundaries have become the dominating part of the system, with the corners being the most informative regions. The boundary region in the complexity field indicates, how far inside the system the diffusion process has approached. A gap in the boundary region can be observed at the center of the upper and lower boundary region. In these parts, the constant gradient on the boundary already has the average value of 0.5. Several dominant regions can be distinguished in the inner part, being relevant up to the final step. The two images of time step 1200 illustrate very well, that not only the propagating front of diffusion is important, but the whole strip starting from the boundary.

6.3 Local Statistical Complexity of CFD Datasets

The diffusion example perfectly matches the requirements of the method, but is rather trivial from an application point of view. When analysing simulations from the field of fluid dynamics, local statistical complexity has to be applied carefully.

Many computational fluid dynamic (CFD) datasets are solutions to the Navier-Stokes Equations for incompressible flow:

$$\frac{\partial}{\partial t}\vec{u}+(\vec{u}\cdot\nabla)\vec{u}+\nabla p = \frac{1}{Re}\Delta\vec{u}+\vec{g} \tag{11}$$

$$\nabla\cdot\vec{u} = 0 \tag{12}$$

where $\vec{u}$ is the velocity, p is the pressure, $\vec{g}$ are body forces, and Re is the Reynolds number [14]. In incompressible flow, density is assumed to be constant, $\rho(\vec{x},t)=\rho_\infty=const$. When simulating a fluid using these equations, pressure is used to ensure, that divergence is 0, i.e., Eq. 12 holds. Therefore, in each iteration of the simulation, the pressure is adapted all over the field, whereby pressure is no longer a local quantity, and $c=\infty$ for the light-cones. Thus, the cones would have to comprise the whole field in order to capture all information affecting the current position. Analyzing only the velocity by means of local statistical complexity, neglects important information, as velocity and pressure are coupled variables. Hence, an exact solution to incompressible flow systems is not possible using the current method. In Sec. 6.5 the complexity fields of velocity, pressure, and of velocity and pressure are analysed. While the investigation of the individual fields is incomplete, the analysis of both fields gives very good approximations of the relevant structures even with small cones ($c=1$) for pressure. An alternative approach is to use the vorticity, which is not influenced by the pressure. Examples of both approaches will be shown in the following two sections in context.

It should be noted that this restriction holds only for incompressible flow simulations. Finite difference simulations of compressible flow fit perfectly the requirements of local statistical complexity analysis.

6.4 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. [6]

$$Re \equiv \frac{v_z(0,z_0)D}{\nu} \qquad S \equiv \frac{2v_\theta(R/2,z_0)}{v_z(0,z_0)} \tag{13}$$

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 [26] for the convective terms. The time integrator is an explicit s-stage, state space Runge-Kutta method ([8], [21]), 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 [8]. The Helmholtz PDE for streamfunction $\tilde{\Psi}(r,z,t)$ is solved with an iterative method using deferred corrections

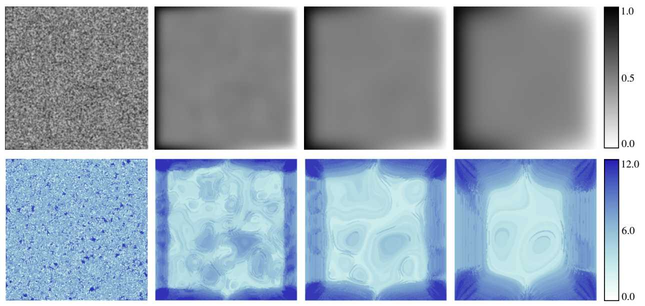


Fig. 3. An isotropic diffusion with fixed concentrations at the boundaries, inducing a gradient from the left to the right handside. The upper row shows the concentrations at time step 2, 200, 500, and 1200. Regions of high importance for the system's dynamics, extracted using local statistical complexity, are displayed in the lower row.

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 $\mathscr{D}=\{(r,z):0\leq r\leq R, 0\leq z\leq L\}$ 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 [6], [27]) 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. 4(top)). Fig. 4(top/left) shows a LIC of the velocity field, featuring several vortices. When overlayed with a transparent map, hiding regions of very low velocity, a better impression of the flow is provided, as many vortices are detected in regions close to noise. Local statistical complexity is computed for velocity and vorticity separately (Fig. 4(bottom/ middle and right)), and for both fields at a time (Fig. 4(bottom/left)). The main structures that have developed up to the instant of the analysis are a conical shear region, outlined in blue in Fig. 4(top/left), and several ringlike vortex structures, one being marked with red points in Fig. 4(top/left). Both features are detected by local statistical complexity. Only minor differences exist between the analysis of velocity, vorticity, and the combination of both fields. This might be due to the fact, that the system's dynamics are quite complex. Thus, it is easier to distinguish between regions, where it is difficult to predict the system's future dynamics, and those where it is easy. As this differentiation is quite clear, it is present in both fields. 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.

6.5 Flow Around a Cylinder

The flow around a cylinder is a widely researched and well understood problem in fluid mechanics. Over a certain range of Reynolds numbers, the flow becomes asymmetric and unsteady, forming the well known von Kármán vortex street [14]. The flow is simulated using NaSt2D, the solver explained in [14], and available online at [10]. The configuration file for a flow around a cylinder provided with the implementation is used for the simulation, ensuring correct settings. The grid consists of 660×120 positions. 4500 time steps are simulated, ranging from times $t=0.0$ to $t=44.53$. The simulation output consists of three fields: velocity, pressure, and vorticity, which are visualized for time step 3402 in Fig. 7.

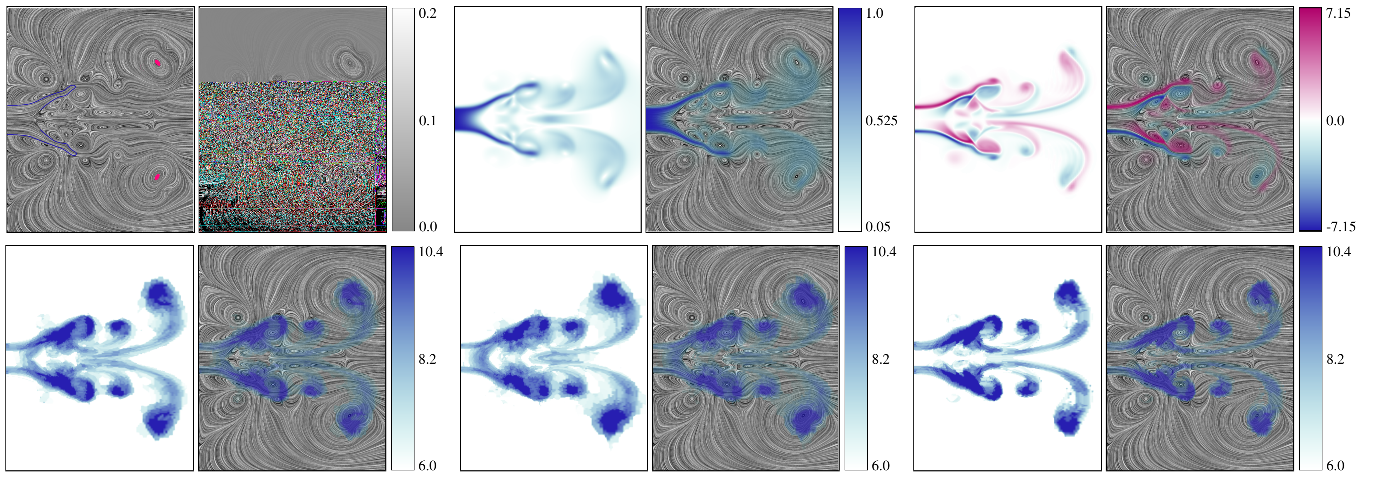


Fig. 4. **(top)** Illustrations of the swirling flow: **(left)** (left)LIC of the velocity of the swirling flow. The conical shear region is outlined in blue. Several ringlike vortex structures can be observed, one being marked by red points. (right) The LIC is overlayed by a transparent mask, hiding regions of small velocity. Thus, the structure of the flow is clarified. **(middle)** Norm of the velocity. **(right)** Vorticity.
(bottom) Local statistical complexity fields of the swirling flow from left to right: velocity and vorticity, velocity, vorticity.

The corresponding local statistical complexity fields are displayed in Fig. 9. As stated earlier, the presented method cannot be directly applied to the pressure. Fig. 9(top) shows the complexity of the pressure, using light-cones with $c = 1$. It is clearly visible, that the analysis lacks information, as the structures are no longer continuous and feature holes. The complexity field of the velocity (Fig. 9(second)), highlights the important structures in the flow. Most dominant is the region, where the flow hits the object, stretching to the seperation structures behind the cylinder. The region where vortices originate is also marked in dark blue, as well as the current positions of vorticies in the vortex street. A ribbon shaped like a sinus runs through the whole vortex street, linking the periodically created vortices. The complex regions at the top and bottom mark regions of shear flow, induced by the boundaries. In general, the complexity field of the vorticity exhibits the same structures. Nevertheless, in both images the structures are not fully developed, as information is missing. Fig. 9(bottom) shows the complexity of all three fields. As more information is included in this analysis, the relevant regions get more dominant and smoother.

The fifth image in Fig. 7 shows the λ_2-field of the current time step. The λ_2 criterion is a standard technique to visualize vortices [32]. A vortex is defined as the region where $\lambda_2 < 0$. Thus, the blue regions in Fig. 7 (bottom) are of relevance, featuring the position of the vortices in the von Kármán vortex street, the regions of separation at both sides of the cylinder, and the shear flow at the boundary. The positions of the vortices match perfectly the regions of highest complexity in the sinus shaped ribbon. The major advantages of λ_2 and vorticity plots are its short computation times (only a few seconds). As long as the flow's dynamic is rather simple, these methods are more advantageous. The well known and simple examples were chosen to prove the correctness of local statistical complexity. To tap its full potential, more complicated flows are needed that are often simulated on unstructured grids that cannot be analysed so far. An advantage of complexity analysis is the fact that it shows smoother structures, providing a good context. The interaction of the flow behind the cylinder is more clearly represented by local statistical complexity, as the system is considered as a unity and not seperated into different classes of behavior, as done by the λ_2-criterion and vorticity.

6.6 Weather and Climate

In order to test the method also with data with more complex statistical properties, we have chosen data from a global climate simulation done by the Max Planck-Institute for Meteorology (MPI-M) at the German Climate Computing Center (DKRZ). The data was acquired from the "World Data Center for Climate" (WDCC) database. Due to time constraints we have selected a subset of only one simulated year and 15 surface variables of the atmospheric component of the coupled atmosphere-ocean-model ECHAM / MPI-OM [31]. With a time resolution of 6 hours the data includes the daily cycle. The spatial resolution is approximately 200 km.

Fig. 7(top) shows the wind and the evaporation of timestep 247. The combination of both fields shows that a visual analysis is rather difficult as the system is quite complex and no simple structures attract attention. When analyzing these two fields with local statistical complexity several dominant regions appear (Fig. 7(bottom)). Timings are given in Table 1. A region well known for its high spatio-temporal variability is the north atlantic area. Storm systems often develop in the western part of the atlantic, and many of them move across the north atlantic towards europe. Storm activity is highly correlated with windspeed and other meteorological parameters like vorticity, precipitation and evaporation. Due to the chaotic behavior of the weather and climate system, the deterministic forecast of storm tracks for a couple of days is still a scientific challenge. Our method highlights especially those features of the weather system which occur irregularly and hence are difficult to forecast.

The video shows the temporal development of wind and evaporation. The diurnal cycle of evaporation is clearly visible. In a long-term analysis using local statistical complexity this regularity should be detected and classified as little complex. As can be seen in the complexity video of the weather, the current method is not capable of filtering these events, as there are too large fluctuations in the daily curves. The detection of regular events will be part of future work.

7 Conclusion and Future Work

In this paper, local statistical complexity is introduced, as a method to detect important regions in systems that can be described by PDEs. Four examples are used to illustrate the new technique, which detects crucial structures in the system automatically. Unlike most feature detection methods, local statistical complexity does not rely on specifications given by the users, but estimates complexity from the system's dynamics. Thus, the technique is perfectly suited to investigate systems that are only little understood, and give hints, where to look for important regions. The interpretation of complexity becomes increasingly difficult as more variables are included. Hence, the complexity analysis of the weather simulation is executed with only two variables. Research towards a better understanding of the interactions of several variables and their effect on the complexity field will be subject of future work.

Acknowledgements

The authors wish to thank the developers of NaST2D for making the program publically available. Special thanks go to Michael Böttinger for providing the weather simulation and for many helpful remarks.

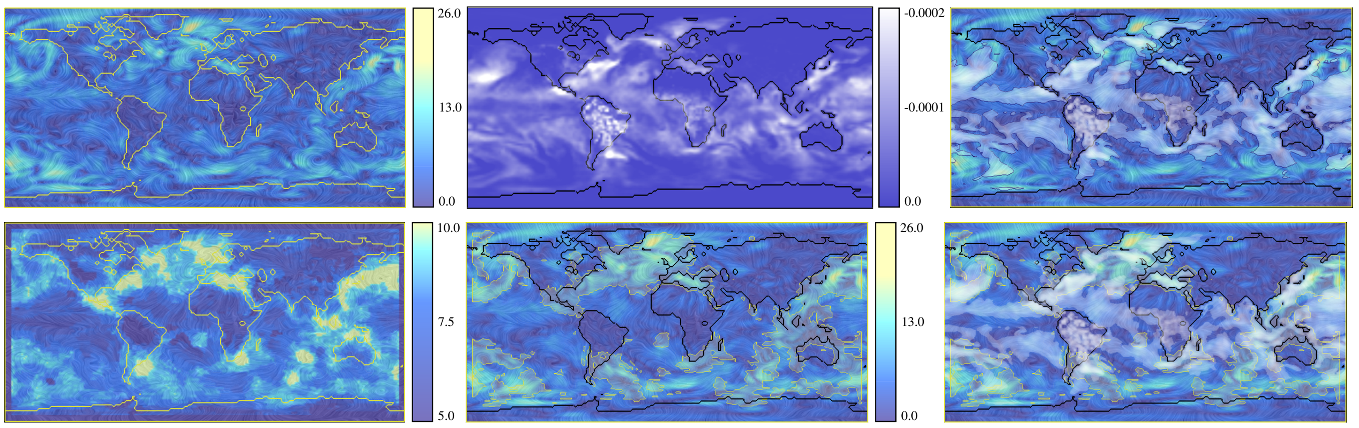


Fig. 5. Simulation of the earth's atmosphere (timestep 247) **(top)** (left) LIC of the wind 10m above the ground. (center) Evaporation. (left) Wind LIC with transparent evaporation overlayed. **(bottom)** Complexity of wind and evaporation (left) as colormap on wind LIC. (center) as highlighted regions (complexity > 8.5) over wind LIC. (right) as highlighted regions (complexity > 8.5) with wind and evaporation.

References

[1] O. Andreassen. Visualization of vector fields using seed LIC and volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):673–682, 2004. Member-Anders Helgeland.

[2] R. Badii and A. Politi. *Complexity: Hierarchical Structures and Scaling in Physics*. Cambridge University Press, Cambridge, 1997.

[3] A. Bair, D. H. House, and C. Ware. Texturing of layered surfaces for optimal viewing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1125–1132, 2006.

[4] P. Ball. *The Self-Made Tapestry: Pattern Formation in Nature*. Oxford University Press, 2001.

[5] W. Bialek, I. Nemenman, and N. Tishby. Predictability, complexity, and learning. *Neural Computation*, 13:2409–2463, 2001.

[6] P. Billant, J. Chomaz, and P. Huerre. Experimental study of vortex breakdown in swirling jets. *JFM*, 376:183–219, 1999.

[7] J. P. Crutchfield and K. Young. Inferring statistical complexity. *Phys. Rev. Lett.*, 63(2):105–108, July 1989.

[8] G. W. E. Hairer. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, 2nd ed.* Springer Verlag, Berlin, 1996.

[9] J. Ebling and G. 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.

[10] ftp://ftp.lrz muenchen.de/pub/science/fluiddynamics/cfd/NaSt2D.

[11] C. Garth, G.-S. Li, X. Tricoche, C. D. Hansen, and H. Hagen. Visualization of coherent structures in transient flow. *to appear in Proceedings of TopoInVis 2007: Topology-Based Methods in Visualization, Leipzig, Germany, March 4-6, 2007.*

[12] P. Grassberger. Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics*, 25(9):907–938, Sept. 1986.

[13] J. M. Greenberg and S. P. Hastings. Spatial patterns for discrete models of diffusion in excitable media. *SIAM Journal on Applied Mathematics*, 34:515–523, 1978.

[14] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical simulation in fluid dynamics: a practical introduction*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[15] H. Gutowitz. A massively parallel cryptosystem based on cellular automata. In B. Schneier, editor, *Applied Cryptography*. K. Reidel, 1993.

[16] H. Hartman and P. Tamayo. Reversible CA and chemical turbulence. *Physica D*, 45:293–306, 1990.

[17] E. Heiberg, T. Ebbers, L. Wigstrom, and M. Karlsson. Three-dimensional flow characterization using vector pattern matching. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):313–319, 2003.

[18] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Statistical Theory: Ch. 3 - Testing Hypotheses*. New York: Houghton Mifflin, 1971.

[19] T. Hogg and B. Huberman. Order, complexity and disorder. *Xerox Palo Alto Research Center (preprint)*, 1985.

[20] B. Jähne. *Digital image processing: concepts, algorithms and scientific applications*. Springer-Verlag, London, UK, 1991.

[21] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. *Appl. Numer. Math.*, 35(3):177–219, 2000.

[22] R. M. Kirby, H. Marmanis, and D. H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In D. Ebert, M. Gross, and B. Hamann, editors, *IEEE Visualization '99*, pages 333–340, San Francisco, 1999.

[23] J. Kniss, P. McCormick, A. McPherson, J. Ahrens, J. Painter, A. Keahey, and C. Hansen. Interactive texture-based volume rendering for large data sets. *IEEE Computer Graphics and Applications*, 21(4):52–61, 2001.

[24] A. Kolmogorov. Logical basis for information theory and probability theory. *IEEE Trans. on Information Theory*, 14(5):662–664, Sept. 1968.

[25] S. Kullback. *Information Theory and Statistics*. Dover Publ., Inc., 1968.

[26] Y. Li. Wavenumber-extended high-order upwind-biased finite-difference schemes for convective scalar transport. *J. Comput. Phys.*, 133(2):235–255, 1997.

[27] H. Liang and T. Maxworthy. An experimental investigation of swirling jets. *Journal of Fluid Mechanics*, 525:115–159, Feb. 2005.

[28] S. Omohundro. Modelling cellular automata with partial differential equations. *Physica D Nonlinear Phenomena*, 10:128–134, Jan. 1984.

[29] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

[30] K. Riley, D. Ebert, C. Hansen, and J. Levit. Visually accurate multi-field weather visualization. In *Proc. of the 14th IEEE Visualization 2003*, pages 279–286, Washington, DC, USA, 2003. IEEE Computer Society.

[31] E. Roeckner et al. *The atmospheric general circulation model ECHAM 5. PART I: Model description*. Tech. Rep. 349, Max Planck Inst. for Meteorol., Hamburg, Germany, 2003.

[32] 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.

[33] N. Sauber, H. Theisel, and H.-P. Seidel. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):917–924, 2006.

[34] G. Scheuermann and X. Tricoche. Topological Methods for Flow Visualization. In C. D. Hansen and C. R. Johnson, editors, *The Visualization Handbook*, pages 341–358. Elsevier, Amsterdam, 2005.

[35] C. R. Shalizi. Optimal nonlinear prediction of random fields on networks. In M. Morvan and É. Rémila, editors, *Discrete Models for Complex Systems, DMCS'03*, volume AB of *DMTCS Proceedings*, pages 11–30. Discrete Mathematics and Theoretical Computer Science, 2003.

[36] C. R. Shalizi, R. Haslinger, J.-B. Rouquier, K. L. Klinkner, and C. Moore. Automatic filters for the detection of coherent structure in spatiotemporal systems. *Physical Review E*, 73:036104, 2006.

[37] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, July 1948.

[38] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In *VIS '99: Proceedings of the conference on Visualization '99*, pages 35–42, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[39] S. Wolfram. *Cellular Automata and Complexity: Collected Papers*. Addison-Wesley, 1994.

[40] P. Wong and R. Bergeron. *Years of Multidimensional Multivariate Visualization*. Scientific Visualization - Overviews, Methodologies and Techniques, IEEE Computer Society Press., 1997.

Table 1. **Timings** for Isotropic Diffusion (Diffusion), Swirling Flow (Swirl), Flow Around a Cylinder (Cylinder), and Weather (weather). The following abbreviations are used: the different *Fields* used for the analysis are vector (v) or scalar (s) valued; the different *Implementations* used are simple (none of the efficient implementation strategies is used), or efficient (all strategies are used); *Past and Future Depth* denote the depth of the past and future light-cones respectively; *# Representatives* is the number of representative used in the classification process; *Size of List* gives the number of candidates in the classification; *# Omitted* denotes the number of time steps being omitted, when classifying the representatives.

Dataset	Fields	Implementation	Past Depth	Future Depth	# Representatives	Size of List	# Time Steps	# Omitted	Time
Cylinder	2s, 1v	simple	3	3	200	-	5	0	1 h 20 min
Cylinder	2s, 1v	efficient	3	3	200	700	5	0	14 min
Cylinder	2s, 1v	efficient	2	2	5000	1	1	0	58 min
Cylinder	2s, 1v	efficient	2	2	5000	700	1	0	12 min
Cylinder	2s, 1v	efficient	2	2	9000	700	300	20	4h 5min
Swirl	1s, 1v	efficient	2	2	5000	600	1	0	6 min
Diffusion	1s	efficient	2	2	5000	600	1	0	4 min
Weather	2s	efficient	3	3	4000	1000	1	0	14 min

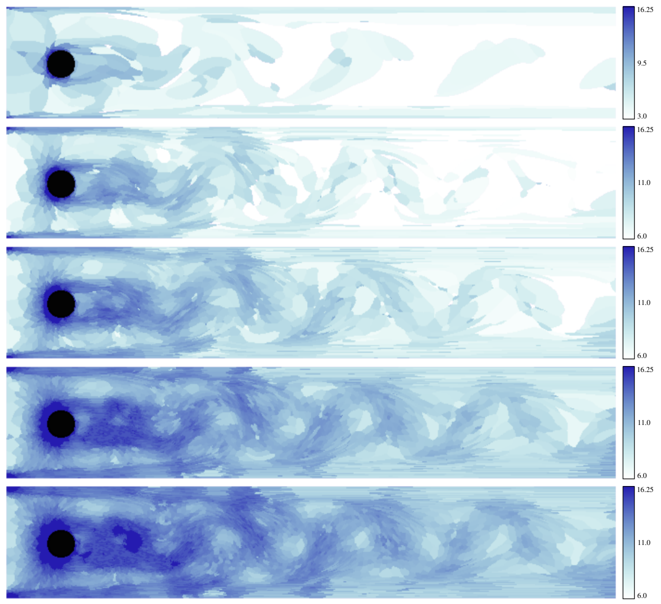


Fig. 6. The complexity field of the flow around a cylinder (time step 3402) with different numbers of representatives. From top to bottom: 200, 1000, 2000, 5000, 8000.

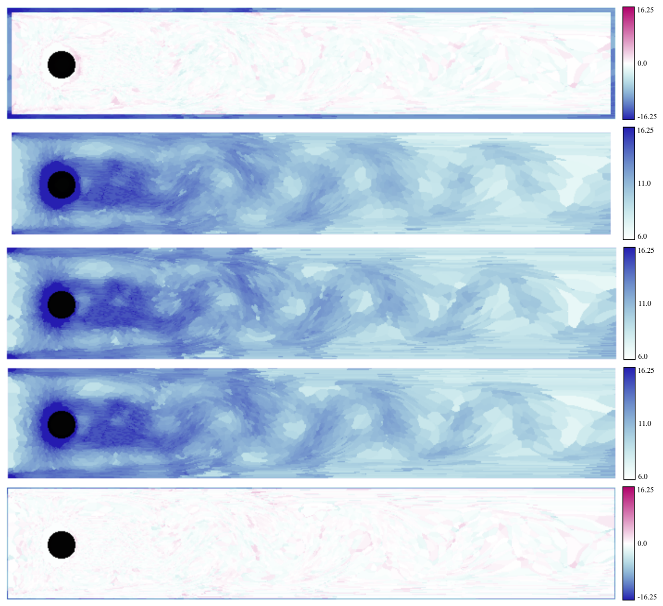


Fig. 8. The complexity field of the flow around a cylinder (time step 3402) with different depths of the cones. From top to bottom (past depth/ future depth): difference between (6/6) and (1/2), complexity for (6/6), complexity for (1/2), complexity for (2/2), difference between (2/2) and (1/2)

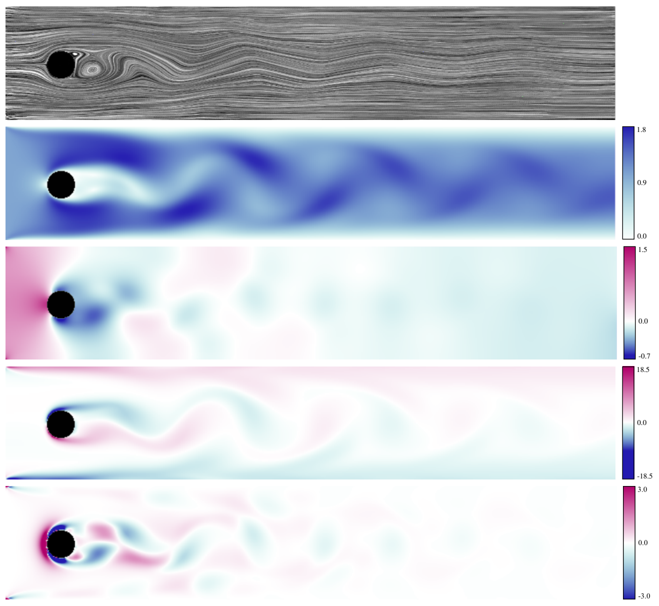


Fig. 7. Flow around a cylinder: The images are a snapshot at time $t = 33.637066$ (time step 3402). The first and second image display the velocity using LIC and a colorcoding of the norm of the velocity. The third image shows pressure, the fourth vorticity, and the fifth λ_2.

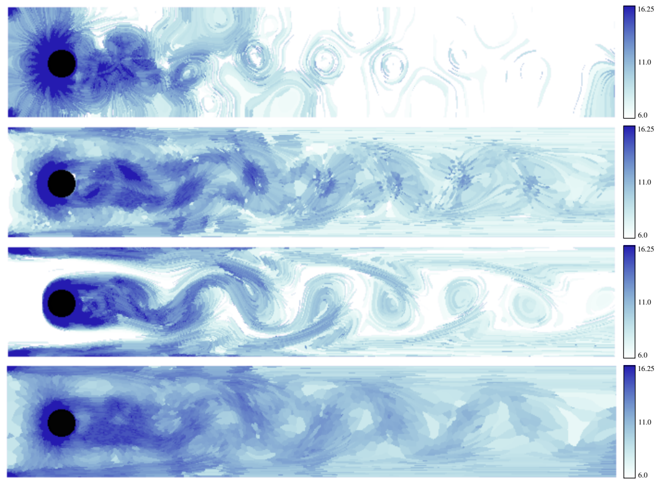


Fig. 9. Complexity fields of the flow around a cylinder (time step 3402) from top to bottom: Pressure, velocity, vorticity, pressure and velocity and vorticity. For all computations parameters are chosen as follows: depth of past and future cones 2, number of representatives in classification 5000, size of candidate list in classification 600, number of time-steps 1