

Automatic Detection and Visualization of Distinctive Structures in 3D Unsteady Multi-fields

Abstract

Current unsteady multi-field simulation data-sets consist of millions of data-points. To efficiently reduce this enormous amount of information, local statistical complexity was recently introduced as a method that identifies distinctive structures using concepts from information theory. Due to high computational costs this method was so far limited to 2D data. In this paper we propose a new strategy for the computation that is substantially faster and allows for a more precise analysis. The bottleneck of the original method is the division of spatio-temporal configurations in the field (light-cones) into different classes of behavior. The new algorithm uses a density-driven voronoi tessellation for this task that more accurately captures the distribution of configurations in the sparsely sampled high-dimensional space. The efficient computation is achieved using structures and algorithms from graph theory. The ability of the method to detect distinctive regions in 3D is illustrated using flow and weather simulations.

Categories and Subject Descriptors (according to ACM CCS): I.6.6 [Simulation and Modeling]: Simulation Output Analysis J.2 [Physical Sciences and Engineering]: Mathematics and Statistics

1. Introduction

Due to the increase in compute power, computer simulations result in increasingly large data sets. Often, the data is 3-dimensional, time dependent and multivariate. Visualization techniques can help reduce the giant amount of information provided by the data to the important facts. An example for the importance of such techniques is aircraft design. To reduce costs large parts of the development and testing of an aircraft are performed using computer simulations. Thus, flight quality and performance can be tested in many different scenarios. In order to analyze the behavior of the aircraft, developers have to extract relevant features from the simulation and visualize them. Relevant structures are for example the wake of an airplane or vortex breakdown. Understanding the wake of an airplane is important to decide at which intervals airplanes can start and land. Planes that enter the wake of a preceding flight might experience large turbulences that can cause a crash. Similarly if the bend that a delta wing flies is too sharp the vortex that keeps it in the air might break down causing a crash. Finding such structures early in the design process is an important task. In most application domains, scientific data visualization is used to understand both the mean features as well as unusual patterns in the data.

A standard technique to detect relevant structures is feature extraction. Post et al. [PVH*03] provide a detailed ex-

amination of techniques that are available in the field of flow visualization. The methods are divided into approaches based on image processing, vector field topology, physical characteristics and selective visualization. For a detailed explanation we refer the interested reader to this publication. Opposed to feature extraction methods that extract single formations, structure-based visualization focuses on a partitioning of the entire domain. Techniques can be classified as cluster-based or integral-line based methods. Salzbrunn et al. [SJWS07] provide a good overview over these approaches. Methods from both fields have in common that they need a priori information about the structures that are to be found. Moreover, most techniques were developed for a special type of field, e.g., scalar or vector valued. Thus, a precise definition of what is to be found is needed that cannot always be given. For example, there exists no general definition of a vortex.

Recently Jänicke et al. [JWSK07] adopted local statistical complexity, a concept from information theory, and used the method to automatically detect structures in an unsteady multi-field that deviate from the average behavior in the field. Due to high computational costs, the method was so far restricted to 2D data-sets. In this paper we propose a new strategy for the computation that is substantially faster and allows for a more precise analysis. In the examples we will focus on individual variables of the multi-fields in order to

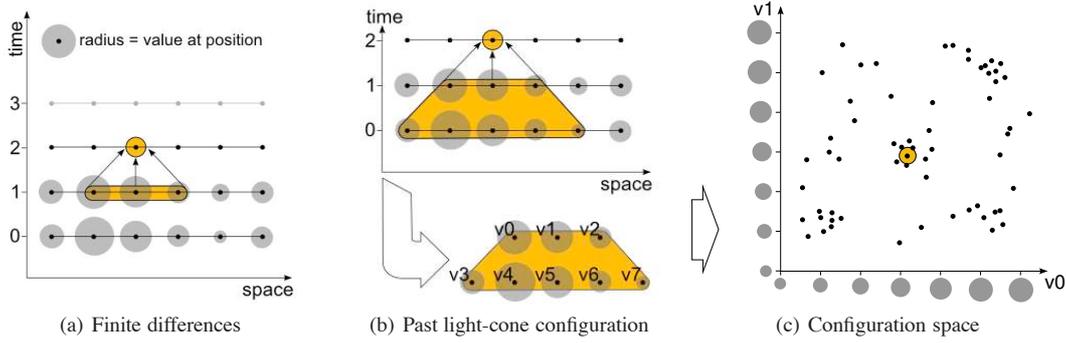


Figure 1: The configuration space: (a) Neighboring values (brown cone) in the previous time-step are used to compute the value at the current position (green). (b-c) Cone-configurations are extracted from the field and stored in the configuration space.

prove the correctness of the method by comparing it to standard techniques and to get a better understanding of what local statistical complexity extracts.

2. 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 to him or her.

As mentioned before, 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. 1(a)). (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 (Fig. 1(b)). 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. 1(a)). 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 optimal code (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 codeword length that is theoretically possible for the given alphabet X . For functions $f(x)$, 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. [SHR*06] 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 distinc-

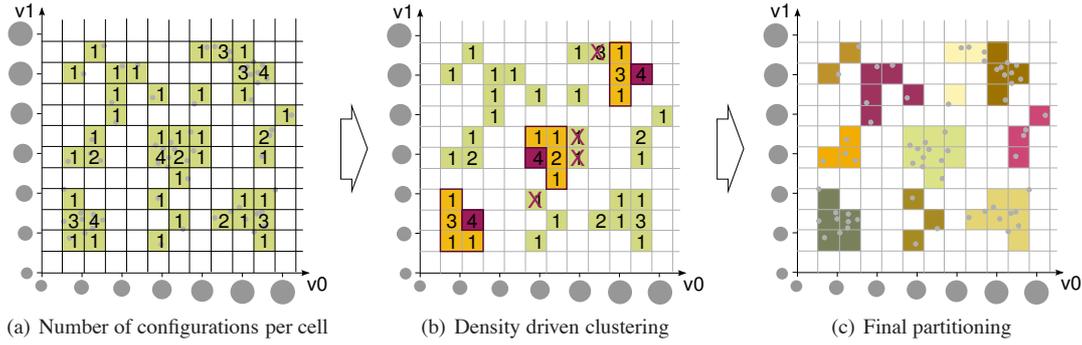


Figure 2: Density-driven voronoi tessellation: (a) The configuration space is quantized first and the number of configurations in each cell are counted. (b) Starting from the cells with most configurations, region growing is performed. (c) The final partitioning captures regions of high density.

tive 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 [SHR*06, JWSK07].

3. Computation of Causal States

3.1. Partitioning of the Configuration Space

As mentioned before, local statistical complexity requires a partitioning of the past-configurations into causal states. For this purpose we examine all configurations from all positions in all time-steps simultaneously in the configuration space (Fig. 1(c)). Note that we treat past and future cones separately. Each entry in the cone configuration is represented in a separate dimension. Hence each configuration is represented as a point in the high-dimensional configuration space. A causal state is a subset of the configuration space. The set of all causal states gives a partitioning of the occupied regions of the configuration space. The partitioning method proposed by Jänicke et al. [JWSK07] extends the ideas of Shalizi et al. [SHR*06] and consists of four steps:

1. Partition the configuration space of past-cones into regions l^- by selecting n^- farthest points.
2. Partition the configuration space of future-cones into regions l^+ by selecting n^+ farthest points.
3. Estimate the conditional distributions $P(L^+|L^- = l^-)$ with $L^+ = \bigcup l^+$ and $L^- = \bigcup l^-$.

4. Cluster those l^- that have similar corresponding distributions $P(L^+|L^- = l^-)$ using a χ^2 -test.

The crucial part are the partitionings in Step 1 and 2. The algorithm proposed by Jänicke et al. [JWSK07] is based on the farthest points method. Their approach, however, has two disadvantages: The computation is very expensive as cone-configurations have to be read from the field very often. Moreover, the farthest points are seeded randomly and thus do not take the distribution of points in the configuration into account. In doing so, clusters of similar configurations might be split and are possibly not joined again by the χ^2 -test as too few samples are present to compute it correctly.

3.2. Density-driven Voronoi Tessellation

With our new method we are able to overcome both problems at a time. We propose a density-driven voronoi tessellation of the configuration space that takes clusters into account and computes each light-cone only once. A graph structure is used to accelerate computations.

Quantization of the Configuration Space: When analyzing 3D fields, we have to cope with millions of data points in a (at least) 27-dimensional configuration space. (Cones in 3D are increasing cubes, minimal size $3 \times 3 \times 3 = 27$) As the space is sparsely sampled, we first extract occupied regions. Using a close meshed quantization of the configuration space, we can precisely locate relevant cells (Fig. 2(a)). In order to compute the number of configurations in each cell, we once sweep through the whole data-set, compute the cone-configuration at each point in each time-step, discretize this configuration and store it in a tree. Each level in the tree represents a dimension of the configuration space and a path from the root to a leaf gives a single configuration. The number of occurrences of individual configurations in the whole data-set are stored in the leaves.

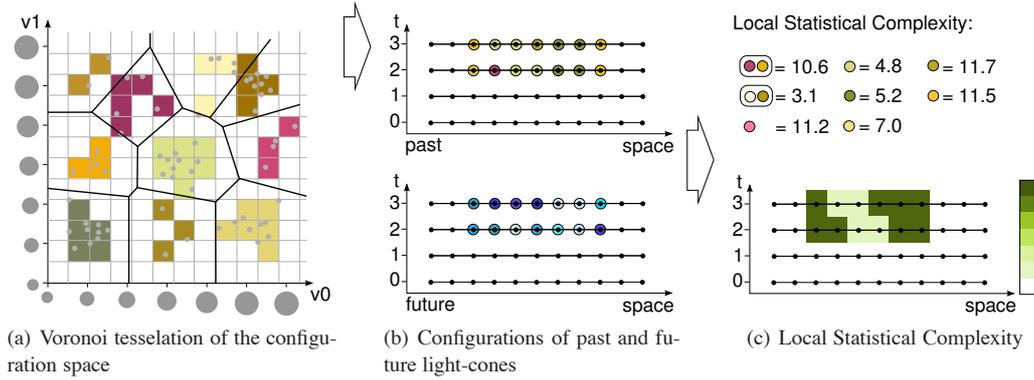


Figure 3: The voronoi tessellation of the past configuration space (a) tells in which class each past-cone configuration in the field is (b-top). The same computation is performed for the future configurations (b-bottom). (c) Afterwards local statistical complexity is computed for each causal state (top) and positions in the past field are marked respectively (bottom).

Region Growing: As explained earlier this simple quantization results in a deficient local statistical complexity. To overcome this problem we compute equally sized voronoi cells that deemphasize the quantization boundaries. First we sort the occupied cells C^i according to their number of samples and store them in a list *NbSamplesList*. We start the region growing at the cell with most samples (Fig. 2(b) red cells). If several cells with the same value exist, we complete them in random order. The distance between two cells is given by:

$$\text{dist}(C^0, C^1) = |C^0 - C^1|^2 = \sum_{k=0}^{\text{coneSize}} |v_k^0 - v_k^1|^2 \quad (1)$$

with *coneSize* being the number of entries v_k in a cone-configuration (compare Fig. 3(b)). Using the tree structure, we can find all cells C^j ($j \neq i$) within distance Δd from C^i (Fig. 2(b) orange cells). C^i and its neighboring cells C^j form a new cluster.

To ensure that the voronoi cells have approximately the same size, we require the centers of the region growing to have mutually minimal distance $2\Delta d$. To clarify this requirement think of a Gaussian distribution in configuration space. The first cell would be centered at the peak and successive cells at its boundary. The boundary cells would capture only half as much space as the central one, and thus have usually fewer hits, which makes them more distinct using local statistical complexity as a measure. To circumvent this problem, cells C^k with distance $\Delta d < \text{dist}(C^k, C^i) < 2\Delta d$ are blocked (Fig. 2(b) red crosses). Blocked cells cannot be used as a center for region growing. For each blocked cell we store the closest center cell C^i and the corresponding distance $\text{dist}(C^k, C^i)$. If in the process of the region growing blocked cells fall into the neighborhood of a new center, they are add to this cell's cluster. The process continues with the next cell in the sorted list *NbSamplesList*. Cells that are already clustered or blocked cannot be used for region

growing and the algorithm directly continues with the next cell. The initial region growing is finished when there are no more cells in the list. Blocked cells that are not assigned to a cluster at the end of the process are assigned to the closest cluster, whose ID is stored in the cell.

Search in High-dimensional Space: The expensive part of the region growing is the detection of neighboring cells in the high-dimensional configuration space. Extensive research has been carried out in this field. The algorithms can be divided in three classes: tree-like index structures, approximate similarity search and indexing by clustering. Li et al. [LCGMW02] give a good overview over the different techniques. We chose a tree-like index structure, as we need exact neighbors and want to find the blocked cells at the same time. Most proposed tree structures like the R*- , SS- or RS-tree (see [LCGMW02] for references) are based on a hierarchical subdivision of the space to reduce the amount of distance computations. These methods are optimized for large databases where the goal is to minimize the number of database accesses. In our case we do not use a database to store the cone-configurations, but have to recompute them from the loaded fields. As this might involve many different time-steps, we would need plenty of swapping to load the required time-steps. Therefore, we choose a different approach, using a standard tree. Each level in the tree represents a dimension of the configuration space and a path from the root to a leaf gives a single configuration. Neighbors and blocked cells can be determined at the same time. The pseudo-code is given in Algorithm 1. The search time can be decreased by storing a label for each node. Nodes that only lead to leaves that are already assigned to a cluster are labeled FINISHED and are no longer considered in following neighborhood requests.

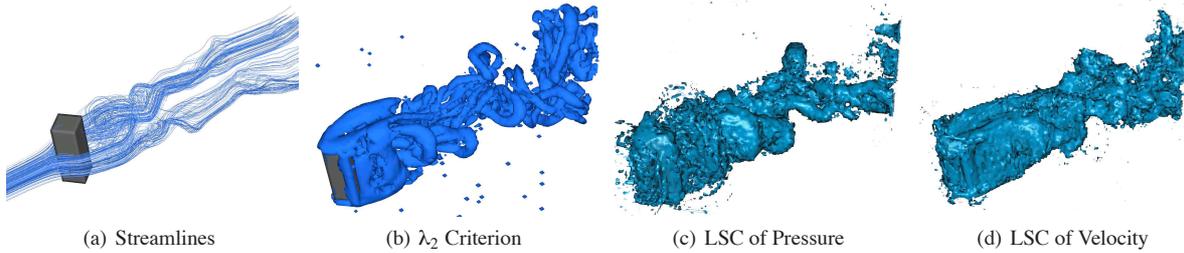


Figure 4: Flow around a cuboid.

3.3. Causal States and Local Statistical Complexity

After the partitioning of the past and future configuration spaces (Fig. 3(b)), the causal states can be determined. For each voronoi cell I^- in the past configuration space we estimate the conditional probability $P(L^+|L^- = I^-)$ with $L^+ = \bigcup I^+$ and $L^- = \bigcup I^-$. Those cells I_i^- that have very similar distributions $P(L^+|L^- = I_i^-)$ are clustered using a χ^2 -test. Each cell in the merged configuration space forms a causal state (Fig. 3(c) top). The local statistical complexity of each causal state is given by:

$$LSC(\text{causalState}) = -\log(P(\text{CausalState} = \text{causalState})) \quad (2)$$

The complexity field is determined by the local statistical complexity of the causal state corresponding to the past-cone at each position (Fig. 3(c) bottom). More details on the theory and implementation of this last step can be found in [JWSK07] and [SHR*06].

4. Results and Discussion

So far we have explained what local statistical complexity theoretically extracts and how it can be computed fast and efficiently. The following three examples are used to illustrate that those structures that are assigned a high local statistical complexity actually are relevant in the context of the data-set. We use examples from flow simulation and a simulation of a hurricane. The first two data-sets are well understood examples from flow visualization that we chose, as we first want to illustrate that local statistical complexity is able to detect those regions that are rated as relevant by experts. The hurricane example is far more complicated and less understood. This data-set is used to explore the capabilities of local statistical complexity.

4.1. Cuboid

The flow around an obstacle is a standard test-case in CFD simulations. In our example, the flow passes a cuboid and evolves into intricate turbulent structures (Fig. 4(a)). In the wake of the cuboid many small vortices interact and form a complex flow pattern. The dataset is simulated on a

100x100x100 rectilinear grid and consists of a velocity and a pressure field.

The local statistical complexity fields of pressure and velocity are very similar (Fig. 4(c), 4(d)). Both pictures show all positions that are assigned a complexity value in the upper third. The resulting structure stretches from the cylinder to the end of the dataset and covers the region of turbulent flow. Different vortices can be identified that leave the main structure and enter again. The coarse resolution is due to the small amount of positions in the data-set, which limit the number of different classes that can be extracted. The complexity field was computed for a single time-step in less than a minute. Integrating additional steps in the analysis did not improve the results significantly, as the dynamics in the flow change very rapidly and similar patterns hardly ever occur again in later time-steps. As vortices are the dominating structures in the data-set, an image showing the vortices is given in Figure 4(b), which displays the λ_2 -criterion [JH95], a standard technique for vortex detection. The individual vortices are much better captured by λ_2 , but essentially we get the same structures. This means that the vortices are the major cause of irregularities in the data-set and that a lot of information is needed to predict the temporal evolution of the elements in a vortex. As explained earlier, a more precise analysis of the vortices could be achieved with more data. Using a finer resolution, we would get more samples and could analyze the data-set at a finer level of detail and extract more subtle structures.

4.2. Delta Wing

The EDELTA dataset 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. 5(a)). With increasing angle 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

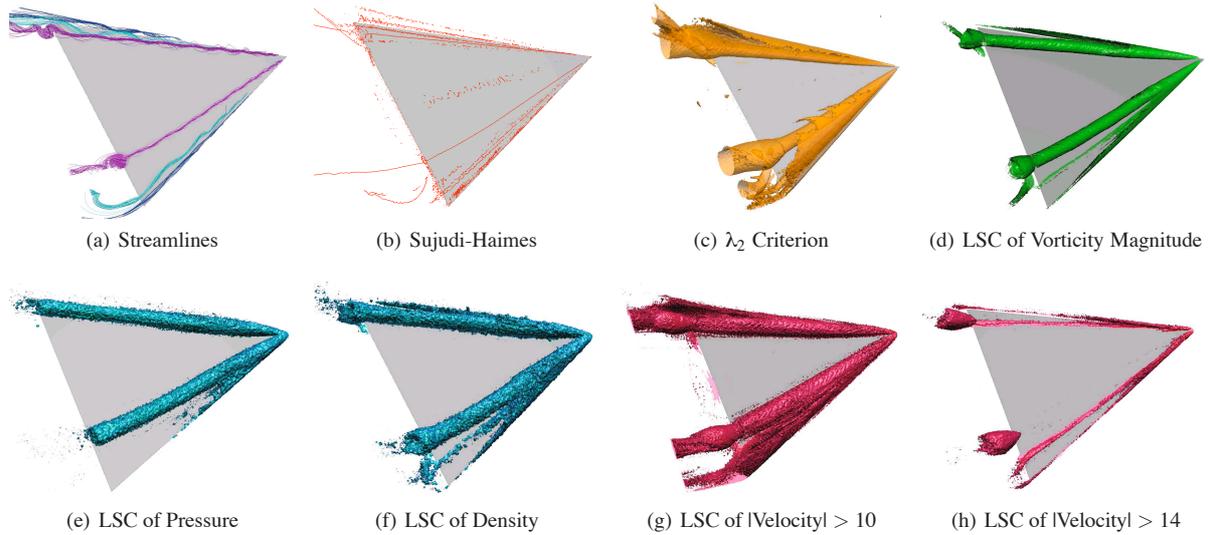


Figure 5: Delta Wing: (a) The interesting structures in the delta wing are two formations of three vortices on either side of the wing illustrated using streamlines in different colors. Recirculating bubbles can be observed at the ends of the primary vortices. (b) Sujudi-Haimes of the vector field. (c) λ_2 -criterion applied to the Jacobian of the velocity field. (d) Local statistical complexity applied to the density field.

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 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 Mio positions) and consists of more than 1000 time-steps. The images in Figure 5 depict time-step 700. The computation of the complexity fields took less than five minutes each for a single time-step with corresponding past and future.

The upper row of Figure 5 gives an overview over standard vortex detection techniques. The algorithm by Sujudi and Haimes [SH95] (Fig. 5(b)) 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 the surface, whose vortices are less dominant and interact with each other. The same observation holds for the λ_2 -criterion, which extracts a “hull” of the vortex, where the two smaller vortices are combined in a single structure. The best separation was achieved by an isosurface of the magnitude of the vorticity.

The lower row of Figure 5 shows the local statistical complexity of different quantities. While the major vortices are well extracted by the complexity field of the pressure (Fig. 5(e)), the smaller vortices and the recirculation bubbles are

barely visible. This means that the local temporal evolution of the pressure in this regions is very similar to ordinary pressure dynamics in the field. A more distinguished representation is provided by the complexity field of the density (Fig. 5(f)). Although all interesting structures feature distinct behavior with respect to the density, the three vortices cannot be separated very well. The most extraordinary dynamics could be observed in the field of the norm of the velocity. Figure 5(g) 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 regions as vortex. Increasing the complexity value further (Fig. 5(h)), we see that the individual vortices are well 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 therefore capable of identifying structures that exhibit an extraordinary formation without precisely describing its pattern.

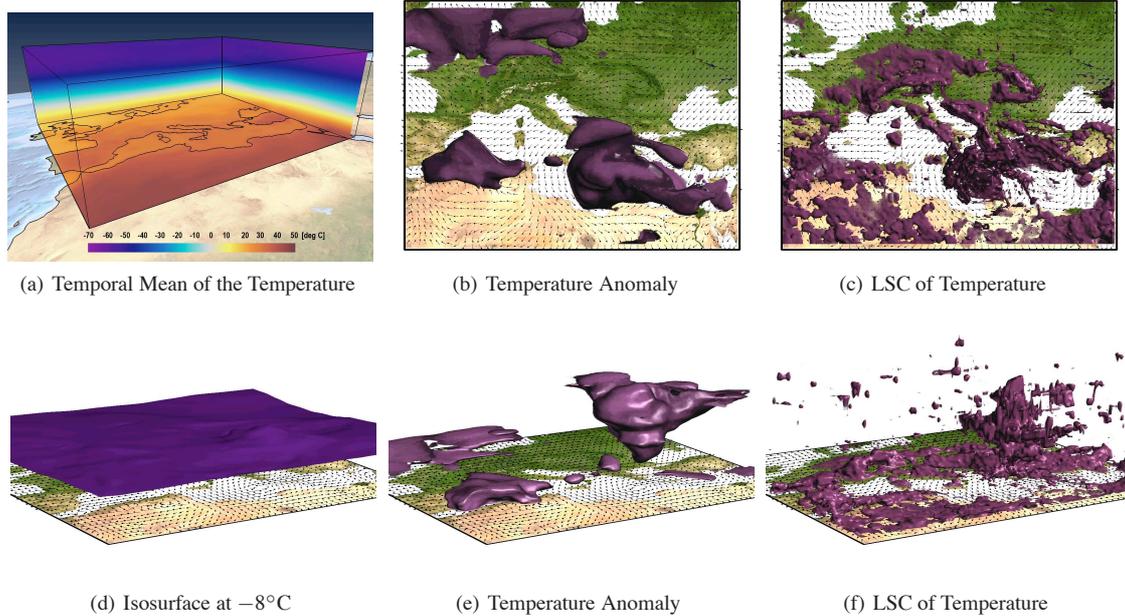


Figure 6: Time-step 60 of the Hurricane Data-set

4.3. Hurricane

In [JWSK07], 2D-LSC has been applied to simulation results from different application domains. The potential ability of the method to detect and enhance exceptional features in multivariate time dependent flow fields was demonstrated using different physical quantities. LSC highlighted specific regions, but especially for the weather data example, the results may have to be further analyzed in order to identify ideal use cases for the method.

In this work we examine the ability of the method to automatically detect the three dimensional structures of regions with anomalous behavior in data where the signal is hidden behind a strong background pattern. For this purpose we selected an atmospheric data set from a warming simulation with a regional climate model. Analyzing results of an ensemble of regional climate models, Gaertner et al. [GJG*07] found a general increase of cyclone intensity over the Mediterranean Sea under a climate change scenario. At least one of the high resolution atmospheric models, REMO, showed even cyclone activity with the characteristic features of tropical cyclones. These are strong and very large low pressure systems which persist for several days. Although strong cyclones can easily be visually detected in some quantities (e.g. sea level pressure), some of the aspects of the cyclone dynamics cannot directly be visualized because of dominating natural patterns - a good test for our method.

If we examine an 3D atmospheric temperature field, we observe a strong vertical temperature gradient ranging from

less than -60°C to more than 40°C . Depending on the geographical extent of the data set, the horizontal temperature variations within a horizontal slice can span more than 50 deg. Figure 6(a) shows a 3 weeks temporal mean of the temperature for late summer using the REMO climate change data. Evidently, the near surface air temperature is higher over land compared to the ocean. Also not surprisingly, the southern regions are generally warmer than the northern countries.

In the time dependent case, pronounced yearly and daily cycles are superimposed to the mean state. Further variations are related to the natural variability of the system, which is dominated by weather patterns.

Outstanding examples of weather induced 3D-temperature anomalies are the patterns connected to cyclone activities: In low pressure systems, warm air rises because of their lower density compared with the surrounding air. The condensation of water vapor releases more energy at higher levels, enhancing the updraft process. Due to this strong convection near the center of cyclones, typical positive temperature anomalies can be observed at higher altitudes.

This time dependent 3D temperature anomaly cannot be visualized directly without applying further processing steps. In order to derive the temperature anomaly pattern, for each grid point a time mean has to be built over an appropriate number of time steps. In our case, we have computed a time mean over a 3 week period in order to reduce the influence the daily cycle and of single weather events.

Algorithm 1 Search_neighbors(double configuration)

```

dist ← allowed distance
nodesToCheck ← (rootNode, remainingDist = 2*dist)
currNode ← nodesToCheck[0]
while nodeToCheck.size > 0 do
  if currNode.hasChildren() then
    for each child that is not FINISHED do
      compute distance to value in configuration
      if distance < currNode.remainingDist then
        nodeToCheck.add( child,
          currNode.remainingDist - distance )
      end if
    end for
  end if
  else
    if currNode.remainingDist < dist then
      for BLOCKED nodes update closest cell
      currNode.state = BLOCKED
    else
      currNode is neighbor
      update FINISHED state
    end if
  end if
  currNode++
end while

```

The original REMO data set has a horizontal resolution of approximately 50 km, defined on a rotated rectilinear grid. The original hybrid vertical model levels (terrain following at the bottom, pressure levels at the top of the atmosphere) have been interpolated onto 30 unevenly distributed height levels, allowing for a higher resolution near the surface. For our purposes, the data was interpolated onto a horizontally regular grid, using a higher resolution of 0.25 Deg in order to maintain the most important structures. The grid finally used for the LSC analysis has a size of 180x134x30 for a total of 76 time steps, resulting in a field of 723600 positions for each scalar value.

Figure 6 shows different representations of the temperature field for time step 60. On the left panels (a) and (d), the temperature isosurface at 265 Kelvin (-8 deg C) is shown in purple. Due to the strong vertical temperature gradient, the positive anomaly at the location of the cyclone can only be suspected in the area above the bump, but the 3D structure is not visible. The isosurface of the positive temperature anomaly (+4°C) is shown in (b) and (e). Clearly, the location of the large cloud of exceptional warm air is correlated with the cyclonic activity.

In 6 (c) and (f) the results of the 3D LSC method are shown. In contrast to the offline computation of the temperature anomaly, where specific knowledge about the general structure of the data and about the dynamical processes hints us to apply methods like the one described above, LSC automatically gives us information about locations of extraor-

inary temporal changes in the temperature field. Due to the pronounced daily cycle in the near surface temperatures and the sparse temporal resolution of the data (6 hourly), LSC produces small scale structures near the surface in addition to the large scale structure at higher altitudes. The near surface structures distract the perception of the signal related to the cyclone dynamics, but even though, especially in the top view the general shape of the LSC isosurface is in good correlation with both the cyclone position and the temperature anomaly pattern. The perspective view shows differences in the upper part of the structure. Here we find the highest LSC values near the center of the storm, in contrast to the broader shape of the temperature anomaly.

5. Conclusion and Future Work

In our paper we explained the idea behind local statistical complexity, proposed an new method for its fast and accurate computation and showed that local statistical complexity can be used to automatically detect relevant structures in unsteady 3D multi-fields.

References

- [GJG*07] GAERTNER M. A., JACOB D., GIL V., DOMÍNGUEZ M., PADORNO E., SÁNCHEZ E., CASTRO M.: Tropical cyclones over the mediterranean sea in climate change simulations. *Geophys. Res. Lett.* 34, L14711 (2007).
- [JH95] JEONG J., HUSSAIN F.: On the identification of a vortex. *J. Fluid Mech.* 285 (1995), 69–94.
- [JWSK07] JÄNICKE H., WIEBEL A., SCHEUERMANN G., KOLLMANN W.: Multifield visualization using local statistical complexity. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1384–1391.
- [LCGMW02] LI C., CHANG E., GARCIA-MOLINA H., WIEDERHOLD G.: Clustering for approximate similarity search in high-dimensional spaces. *IEEE Transactions on Knowledge and Data Engineering* 14, 4 (2002), 792–808.
- [LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F., D. W.: The State of the Art in Flow Visualization: Dense and Texture Based Techniques. *Computer Graphics Forum* 23, 2 (2004), 203 – 221.
- [PVH*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The state of the art in flow visualization: Feature extraction and tracking. *Computer Graphics Forum* 22, 4 (2003), 775–792.
- [SH95] SUJUDI D., HAIMES R.: Identification of swirling flow in 3d vector fields. *AIAA 95-1715* (1995).
- [SHR*06] SHALIZI C. R., HASLINGER R., ROUQUIER J.-B., KLINKNER K. L., MOORE C.: Automatic filters for the detection of coherent structures in spatiotemporal systems. *Physical Review E* 73 (2006).

[SJWS07] SALZBRUNN T., JÄNICKE H., WISCHGOLL
T., SCHEUERMANN G.: The State of the Art in Flow
Visualization: Structure Based Techniques. *Under review*
(2007).