

# Visualization of Temporal Similarity in Field Data

Steffen Frey, Filip Sadlo, *Member, IEEE*, and Thomas Ertl, *Member, IEEE*

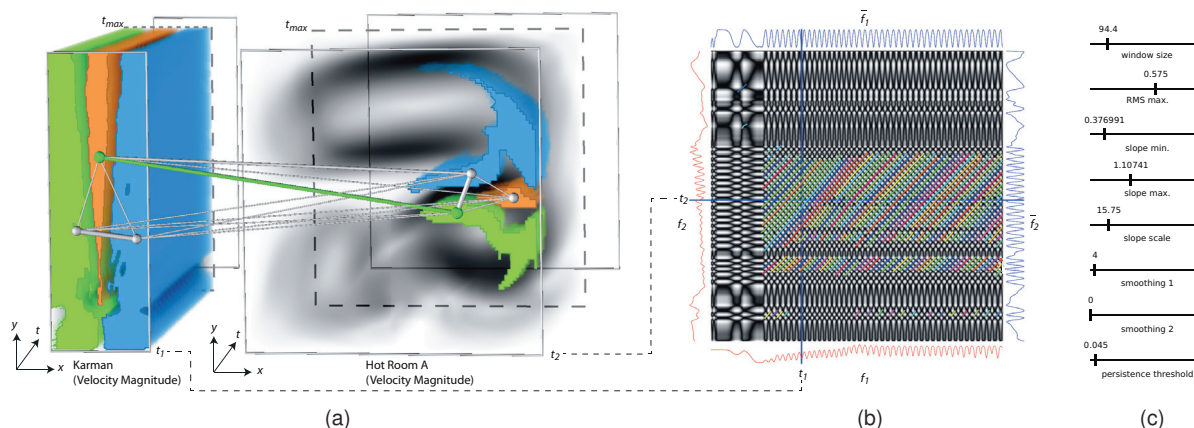


Fig. 1. (a) Space-time view of the time-dependent von Kármán (left) and Hot Room A (right) data sets, with space-time similarity clusters (blue, green, and orange, in order of decreasing size). Temporal similarity between cluster masters (spheres) is depicted by thickness of their links (tubes). One can be interactively selected (green), and (b) the temporal similarity of the respective cluster pair (of the smoothed and normalized signals  $\bar{f}_1$  and  $\bar{f}_2$  at their masters) is visualized by a similarity matrix (gray level plot). (c) The user can interactively parametrize the extraction of temporal similarity from the matrix in terms of similarity lines (colored lines in (b)). The clusters in (a) are obtained on the basis of these lines extracted from similarity matrices, i.e., by utilizing (b).

**Abstract**—This paper presents a visualization approach for detecting and exploring similarity in the temporal variation of field data. We provide an interactive technique for extracting correlations from *similarity matrices* which capture temporal similarity of univariate functions. We make use of the concept to extract periodic and quasiperiodic behavior at single (spatial) points as well as similarity between different locations within a field and also between different data sets. The obtained correlations are utilized for visual exploration of both temporal and spatial relationships in terms of temporal similarity. Our entire pipeline offers visual interaction and inspection, allowing for the flexibility that in particular time-dependent data analysis techniques require. We demonstrate the utility and versatility of our approach by applying our implementation to data from both simulation and measurement.

**Index Terms**—Time-dependent fields, similarity analysis, interactive recurrence analysis, comparative visualization.

## 1 INTRODUCTION

Large parts of science and engineering deal with time-dependent phenomena. While some converge to quasi-stationary behavior after an initial phase of temporal change, others attain periodic or quasiperiodic behavior, or even stay chaotic. There is a multitude of well-established analysis methods with respect to periodic behavior, in particular those operating in the frequency domain of univariate functions. However, many phenomena are beyond strictly periodic behavior and hence not amenable to analysis by these techniques. Nevertheless, as stated in Poincaré’s *recurrence theorem*, they typically exhibit behavior that is characterized by arbitrarily close, but not exact, repetition. In addition, many problems involve multivariate data, i.e., data present as fields parametrized by additional variables such as space.

While it is not uncommon that processes return to a previous state to arbitrary precision eventually, the intent of our work is to reveal similar *sequences* of states, or in other words, similar *processes*. We not only focus on similarity with respect to time alone, but also include temporal similarity between different locations (clustering) and even

different data, providing means for classification based on temporal similarity and *comparative visualization* of time-dependent fields.

Our method is based on the *self-similarity matrix* and *cross-similarity matrix* concepts. These take as input one ( $f_2(t) := f_1(t)$ ) or two ( $f_1(t), f_2(t)$ ) scalar function(s) of time and plot the similarity between their different states in time as a dense gray level 2D plot where both abscissa and ordinate represent the same time interval (Fig. 2(a)). Dark values represent high similarity of the smoothed and normalized signals  $\bar{f}_1$  and  $\bar{f}_2$  (Sec. 4.2). Since similar processes represent similar sequences of states, they appear as dark lines in these plots, which can be geometrically extracted therefrom and are denoted as *similarity lines* (Fig. 2(b)). These lines represent similarity in the temporal variation of the signals (their properties are discussed in detail in Sec. 3.1), e.g., their slope shows the temporal scale of similarity (or the “ratio of their frequencies”). The lines serve as a basis for detection (Sec. 4) and visualization (Sec. 5) of temporal similarity, including the spatial clustering (Sec. 5.4) shown in Fig. 1(a). In practical applications, the lines are filtered, as shown in Fig. 1(b)), according to Sec. 4.5, which influences the overall visualization including the clustering.

A particularly useful property of similarity matrices is their independence of the dimension of the input data, since similarity, as defined in Sec. 4.2, is always a scalar. Hence, these matrices are, irrespective of the data, always 2D scalar representations. Despite the large body of literature on the topic in dynamical systems theory and related fields, to the best of our knowledge, no satisfying technique exists for extracting the lines from similarity matrices. Furthermore, the concepts are basically restricted to individual univariate functions of time. In our

- Steffen Frey, Filip Sadlo and Thomas Ertl are with the Visualization Research Center (VISUS), University of Stuttgart, Germany, e-mail: {steffen.frey, filip.sadlo, thomas.ertl}@visus.uni-stuttgart.de.

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.

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

work we provide a technique for extracting the similarity lines appropriately and present interactive analysis of 2D fields by means of this family of concepts.

As a linked view, we provide the time-dependent fields in space-time representation (Fig. 1(a)). This allows for temporal exploration of the data, i.e., shifting the fields along the time axis and clipping them at the front plane at times  $t_1$  and  $t_2$  to reveal the respective state of the field (see also the dashed end of time indicators ( $t_{\max}$ ) in Fig. 1(a)). At the same time it provides a visualization of the clusters and allows for their interactive exploration, including the similarity matrix view.

In contrast to machine learning approaches, which could also be utilized for these purposes, our entire pipeline supports visual inspection and interaction—from the choice of the underlying comparison function over detection of processes in the similarity matrices, to the analysis of their spatio-temporal context. We make use of linked views and focus+context elements, and for representation of time and browsing, we employ a space-time representation.

We demonstrate the utility of our approach with 2D time-dependent scalar fields, but extension to vector-valued fields is straightforward by substituting the comparison function in our scheme with an appropriate comparison function for vectors. Generally, our technique can easily be adapted to the problem under investigation by replacing the comparison function with an appropriate (domain specific) alternative. Specifically, our contributions include:

- A generic approach to reveal temporal similarity in field data, based on similarity matrices. Sequences of similar states are present in these matrices as lines of locally higher similarity.
- A technique for subpixel-accurate extraction of these lines from similarity matrices, based on a modified marching squares approach and allowing for intersecting lines.
- Visualization of spatial correlation of temporal variation between different parts of fields or data sets.
- Comprehensive visual analysis pipeline providing interactive control during the entire procedure.

This paper is organized as follows: Sec. 2 covers previous work. The fundamentals of similarity matrices are given in Sec. 3. Detection and visualization of similarity are discussed in Sec. 4 and 5, respectively. Sec. 6 presents results and Sec. 7 concludes our work.

## 2 RELATED WORK

Machine learning is an automatic alternative for the analysis of (time-dependent) data, and in visualization particularly interesting with large data [25]. Guo et al. [13] present a system for revealing space-time and multivariate patterns based on self-organizing maps. Berger et al. [4] employ methods from statistical learning to achieve interactive, prediction-based local analysis of continuous parameter space.

While machine learning can take place unsupervised or supervised, there are alternative techniques that make more intense use of human interaction, in particular those utilizing visual data exploration techniques. For example, Rhyne et al. [34] discuss such techniques for geospatial data. Köthur et al. [19] employ clustering and evaluate their approach by means of a data set that is similar to the one we use in Sec. 6.3. Jänicke and Scheuermann [15] use  $\epsilon$ -machines to visualize dynamics in world temperature amongst others.

Kehrer et al. [17] present a system for generating promising hypotheses based on visual data exploration. A visual exploration approach to study climate variability changes on the basis of wavelets was introduced by Jänicke et al. [14]. Fuchs et al. [10] propose an approach combining knowledge-based analysis and automatic hypothesis generation. Lampe et al. [21] introduce a technique based on kernel density estimation that produces expressive pictures revealing frequency information about one or multiple curves. Lee et al. [22] visualize multivariate time-varying data sets based on trend relationships. Kehrer et al. [16] systematically study opportunities for the interactive visual analysis of multi-dimensional scientific data.

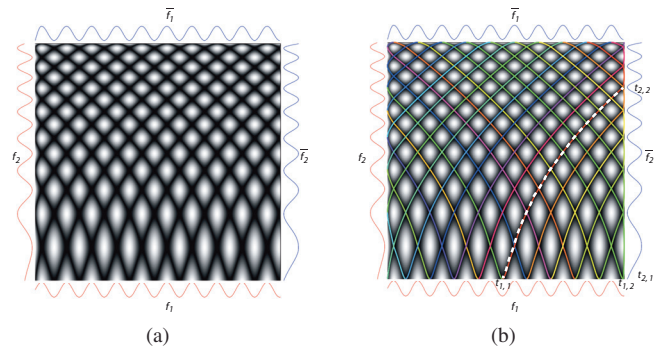


Fig. 2. (a) Similarity matrix (black  $\simeq$  similar, white  $\simeq$  different) and (b) extracted similarity lines. Red graphs (left and bottom of the matrix) show the smoothed signals  $f_1$  and  $f_2$  of  $\sin(x)$  and  $\sin(0.01 \cdot x^2 + 0.1 \cdot x)$ , respectively. Blue graphs (right and top) depict their normalized versions  $\bar{f}_1$  and  $\bar{f}_2$  used for similarity matrix computation.

Gerber et al. [11] utilize topological and geometrical methods for the analysis of high-dimensional scalar fields. Pham et al. [32] present a non-spatial visualization approach of diversity in large multivariate data, while Glatter et al. [12] use textual pattern matching for visualizing temporal patterns. For the visualization both in space and time, Kristensson et al. [20] evaluate and approve the utility of the space time cube representation of time-dependent data.

In path line based time-dependent flow visualization, Salzbrunn et al. [36] present path line predicates, while Sadlo and Peikert [35] introduce a method for analyzing vortex dynamics. Recurrent flow behavior in vortices was addressed by Peikert and Sadlo [30]. Sanderson et al. [37] describe a technique to analyze recurrent patterns in toroidal magnetic fields. An importance-driven visualization of time-varying volume data is presented by Wang et al. [40]. Fang et al. [9] explore time-varying volumetric medical images using time-activity curves (TAC). TACs are also used by Lee et al. [23] to visualize the similarity between a voxel's time series and time-varying features.

In dynamical systems theory, there are the concepts of recurrence plots and similarity matrices for univariate data. We build on those and discuss them in Sec. 3. Their benefits and utility are discussed in detail by Marwan et al. [27]. Vascoceles et al. [39] extend those concepts to multivariate data by breaking down the data into one-dimensional data series and applying recurrence analysis separately. Alternatively, time can be extended into space [28] at the cost of high-dimensional domains, e.g. a time-dependent 2D image is mapped to a 4D recurrence plot, which is, however, hard to visualize. Cutler and Davis [6] utilize recurrence plots for analyzing videos by performing recurrence analysis on segmented objects. Angus et al. [2] introduce conceptual recurrence plots to visualize the similarity of utterances in human discourse. Kononov [18] provides software to render recurrence plots. Bautista et al. [3] analyze the difference between recurrence plots.

Comparative visualization aims to reveal the similarity/difference between data sets, e.g., for verifying scientific simulation codes [1]. Waser et al. [41] introduce an interactive visualization that provides complete control over multiple simulation runs. Sauber et al. [38] discuss a graph-based approach to visualize correlations in 3D multifield data. Malik et al. [26] carry out parameter studies of data set series.

However, to the best of our knowledge, none of these approaches addressed the visualization of temporal similarity in time-dependent *fields*, which allows control and insight by visual data analysis in space-time, comparable to our approach.

## 3 FUNDAMENTALS

The concepts of self-similarity matrices and cross-similarity matrices come from dynamical systems theory. There, the *phase space*  $\Omega \subset \mathbb{R}^n$  represents the set of all possible states of a system and the *phase space trajectory*  $\mathbf{f}: t \mapsto \Omega$  of a system with  $t \in \mathbb{R}$  its behavior over time  $t$ , i.e., its sequence of states in time. According to Poincaré's recurrence theorem many systems return arbitrarily close to their initial (or a previous)

state  $\mathbf{f}(t_i)$  after sufficiently long time  $\Delta t$ , i.e.,

$$|\sigma(\mathbf{f}(t_i), \mathbf{f}(t_j))| < \varepsilon, \quad (1)$$

with  $t_j = t_i + \Delta t$  and comparison function  $\sigma(\cdot, \cdot)$ , traditionally  $\sigma(\mathbf{a}, \mathbf{b}) := \|\mathbf{b} - \mathbf{a}\|$ . This gave rise to the *recurrence plot* concept that encodes the recurrence property of a system over duration  $\tau$ . It is a 2D plot  $\mathcal{R} : \tau \times \tau \mapsto \{0, 1\}$  with 1 at  $(t_i, t_j)$  if Eq. 1 holds and 0 otherwise.

A difficulty with this concept is the appropriate choice of  $\varepsilon$ . Too small values miss recurrences whereas too large ones lead to insignificant representation and will include even consecutive states along the trajectory. The choice of  $\varepsilon$  depends strongly on the system under consideration [27]. An alternative concept that does not require such a parameter is the *self-similarity matrix*  $\mathcal{S} : \tau \times \tau \mapsto \mathbb{R}$  with

$$\mathcal{S}_{ij} = \sigma(\mathbf{f}(t_i), \mathbf{f}(t_j)). \quad (2)$$

A multitude of distance or similarity functions  $\sigma(\cdot, \cdot)$  can be applied in this context. Throughout this paper we use a signed similarity measure, i.e., Eq. 5, but use its modulus for displaying gray level similarity matrices. Note that the respective recurrence plots can simply be obtained from  $\mathcal{S}$  by a threshold operation.

Finally, the extension of the concept to two systems  $\mathbf{f}, \mathbf{g} : t \mapsto \Omega$  yields the *cross-similarity matrix*  $\mathcal{C} : \tau \times \tau \mapsto \mathbb{R}$  with

$$\mathcal{C}_{ij} = \sigma(\mathbf{f}(t_i), \mathbf{g}(t_j)). \quad (3)$$

In contrast to  $\mathcal{R}$  and  $\mathcal{S}$ ,  $\mathcal{C}$  can be non-square because two different systems (signals) are compared, with possibly different time ranges (e.g., Fig. 2(a)). For additional details on the overall topic, we refer the reader to [27]. In our approach, we build on  $\mathcal{S}$  and  $\mathcal{C}$  only.

### 3.1 Properties of $\mathcal{R}$ , $\mathcal{S}$ , and $\mathcal{C}$

Recurrence quantification analysis (RQA) [42, 27] provides measures for a quantitative analysis of  $\mathcal{R}$  based on their small-scale structure. These measures are based on the recurrence point density and the diagonal and vertical line structures of  $\mathcal{R}$ . Since  $\mathcal{R}$  can be obtained by a threshold operation from  $\mathcal{S}$ , these measures apply to  $\mathcal{S}$ , too—and to some extent to  $\mathcal{C}$ . We constrain the description here to the properties and measures that are needed in the context of our technique.

Since we focus on temporal similarity in the sense of similar sequences of states, the feature central to our approach are lines in these matrices, which we call *similarity lines*. Only non-splitting processes are considered in this work, i.e., the lines are not allowed to branch. The matrices  $\mathcal{R}$  and  $\mathcal{S}$  represent the same system in both dimensions, which leads to the so-called line of identity, consisting of the diagonal elements. The properties of the similarity lines have direct interpretations (explained by means of the white-dashed line in Fig. 2(b)):

The *length* of a similarity line relates to the time the phase space trajectory  $\mathbf{f}(t)$  runs parallel to itself (in case of  $\mathcal{R}$  or  $\mathcal{S}$ ) or parallel to  $\mathbf{g}(t)$  (in case of  $\mathcal{C}$ ). In other words, it measures the amount of similar consecutive states. The *bounding box* of a similarity line gives the intervals of temporal similarity in time. The two time intervals spanning the bounding box represent the respective time ranges, and their proportion gives a notion of the overall relative time scaling under which the two sequences are similar. For example, the dashed line in Fig. 2(b) covers the time interval from  $t_{1,1}$  to  $t_{1,2}$  for  $f_1$  and from  $t_{2,1}$  to  $t_{2,2}$  for  $f_2$ . The *slope* at a given time pair of a similarity line reflects the respective relative time scale of similarity. For periodic signals, this represents the relative frequency. For instance, the dashed line has a high slope close to  $(t_{1,1}, t_{2,1})$ , because the frequency of  $f_1$  is much higher there than that of  $f_2$ . Close to  $(t_{1,2}, t_{2,2})$  the slope is approximately  $\pi/4$ , i.e., the frequencies are similar. The second derivative of a similarity line (its *curvature*) reflects the change of temporal scale of similarity. For periodic signals, this represents the relative frequency change. In the example, the frequency of  $f_1$  is constant while the frequency of  $f_2$  increases, reflected by the varying slope along the line. The horizontal (or vertical) *offset* between two lines indicates the phase offset of the respective processes.

There is, however, a prevalent difficulty with the RQA: it is non-trivial to extract the lines from  $\mathcal{R}$ ,  $\mathcal{S}$ , or  $\mathcal{C}$ . Traditional approaches

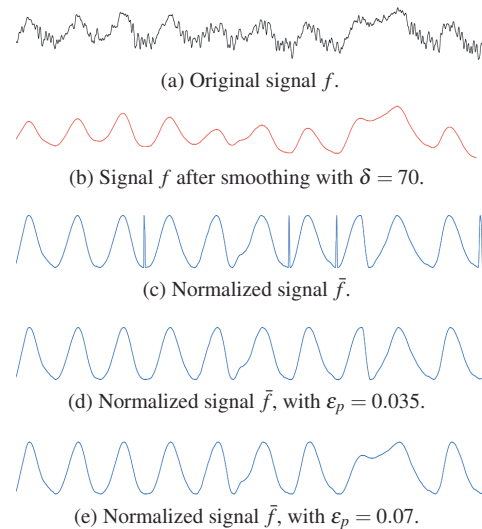


Fig. 3. Signal from a fixed spatial point of the ocean temperature data set (introduced in Sec. 6 in detail). Black depicts the original signal, the smoothed signal is drawn in red and the additionally normalized signals using different persistence thresholds  $\varepsilon_p$  are depicted in blue.

are based on a thresholding operation with subsequent line extraction, suffering from noise and discontinuities. To this end we present a technique in Sec. 4.4 that achieves good results with an alternative formulation of  $\mathcal{S}$  and  $\mathcal{C}$ , as presented in Sec. 4.3.

## 4 DETECTION OF TEMPORAL SIMILARITY

Each element  $(i, j)$  in a similarity matrix represents the similarity of the state, or the data, at the respective times  $t_i$  and  $t_j$ . There is a multitude of similarity measures, in different disciplines and in different applications, each tailored to specific goals and the type of data. It is not our aim here to come up with a new similarity measure—we provide a framework that operates on the basis of an arbitrary similarity measure, or comparison function,  $\sigma : (\mathbb{R}^m, \mathbb{R}^m) \mapsto \mathbb{R}$ . This function compares the two states  $\mathbf{f}(t_i)$  and  $\mathbf{f}(t_j)$  of the time-dependent signal  $\mathbf{f} : \mathbb{R} \mapsto \mathbb{R}^m$ . We demonstrate our approach with scalar data, hence  $m = 1$  and  $\mathbf{f} = f$  in our applications. Nevertheless, since only the comparison function depends on  $m$ , applying our technique to  $n$ -vector data is straightforward.

### 4.1 Data

Although we treat multivariate time-dependent data, i.e., time-dependent  $nD$  fields  $f(\mathbf{x}, t)$  with  $\mathbf{x} \in \mathbb{R}^n$  and time  $t \in \mathbb{R}$ , the comparison function itself operates on points therein, hence we have a time-dependent signal  $f_{\mathbf{x}_i}(t) := f(\mathbf{x}_i, t)$  at each point  $\mathbf{x}_i$  in the spatial domain. In case of flow data, our approach provides the choice to perform similarity visualization either in the Eulerian frame where  $\mathbf{x}_i$  is static and  $f_{\mathbf{x}_i}(t) := f(\mathbf{x}_i, t)$ , or in the Lagrangian frame where  $\mathbf{x}_i(t)$  moves along a path line and  $f_{\mathbf{x}_i}(t) := f(\mathbf{x}_i(t), t)$ . However, since appropriate evaluation of the Lagrangian approach would go beyond the scope of this paper, we address it as future work.

In general, we visualize similarity between two signals  $f_1(t)$  and  $f_2(t)$ . In the traditional self-similarity case,  $f_1(t) = f_2(t) := f_{\mathbf{x}_i}(t)$ . When comparing different points in a field  $f_1(t) := f_{\mathbf{x}_1}(t)$  and  $f_2(t) := f_{\mathbf{x}_2}(t)$ . Finally, if two different fields  $f(\mathbf{x}, t)$  and  $g(\mathbf{x}, t)$  are compared,  $f_1(t) := f_{\mathbf{x}_1}(t)$  and  $f_2(t) := g_{\mathbf{x}_2}(t)$ .

### 4.2 Comparison Function

A comparison function that performed well in our experiments with scalar data applies local normalization to the time-dependent signals  $f_1(t)$  and  $f_2(t)$ . The motivation is to make the comparison invariant to bias and amplitude scale. While global normalization of  $f_1(t)$  and  $f_2(t)$  over the complete time domain would render two signals similar

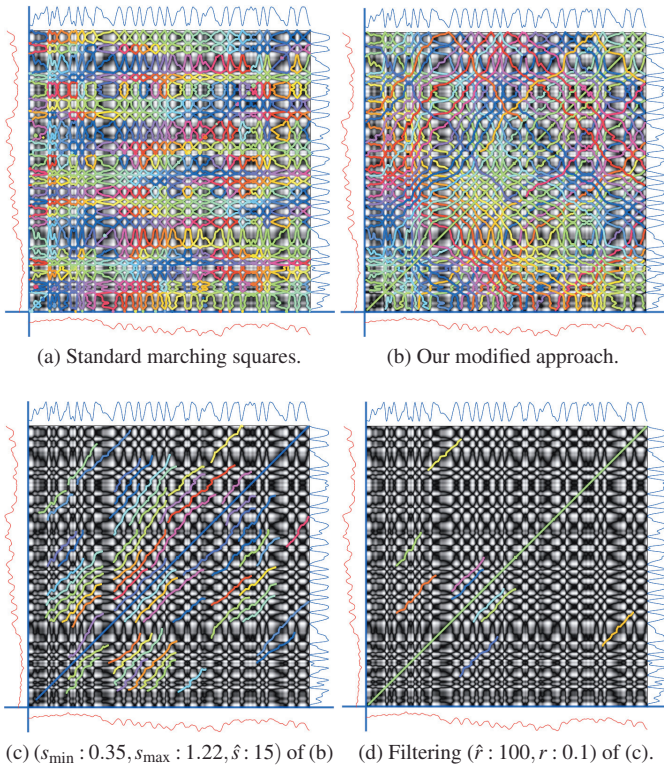


Fig. 4. Similarity lines from a time interval of the function from Fig. 3.

only if they exhibit strongly similar behavior over large intervals, local normalization compensates for drift and change in amplitude. We base the normalization of  $f(t)$  ( $f_1(t)$  and  $f_2(t)$  are normalized independently) on homological persistence [8, 33] of their local extrema. First we apply a smoothing step using a hat kernel of *smoothing kernel size*  $\delta$ . This allows us to select the appropriate time scale (Fig. 3(b)). Then we extract all local extrema  $\hat{t}_i$  within the time domain  $[t_{\min}, t_{\max}]$  and apply topological simplification by removing all persistence pairs  $(\hat{t}_i, \hat{t}_j)$  where  $|f(\hat{t}_i) - f(\hat{t}_j)| < \varepsilon_p$  with  $\varepsilon_p$  being the *persistence threshold*. The remaining  $n \geq 0$  extrema partition the time domain into  $n - 1$  intervals of type  $\tau_i = [\hat{t}_i, \hat{t}_{i+1}]$ , with bounds

$$f(t) \in [f(\hat{t}_i), f(\hat{t}_{i+1})], \quad \forall t \in \tau_i, \quad (4)$$

and possibly two additional intervals  $\tau_{\min} = [t_{\min}, \hat{t}_1]$  and  $\tau_{\max} = [\hat{t}_n, t_{\max}]$ . On the other hand, if  $n = 0$ , there is only a single interval  $\tau_{\min} = [t_{\min}, t_{\max}]$ . Within the  $\tau_i$ ,  $f$  is simply normalized to  $\bar{f}$  by mapping  $\min(f(\hat{t}_i), f(\hat{t}_{i+1}))$  to zero and  $\max(f(\hat{t}_i), f(\hat{t}_{i+1}))$  to one. In contrast to the  $\tau_i$ ,  $f$  is not bounded within  $\tau_{\min}$  and  $\tau_{\max}$  by the values at their boundaries (Eq. 4), hence we step through the values in these intervals to obtain the minimum and maximum and normalize  $f$  therein by mapping these to 0 and 1, respectively (Fig. 3(c)–(e) and Fig. 6(a)–(b)). Using the resulting normalized signals  $\bar{f}_1(t)$  and  $\bar{f}_2(t)$ , our comparison function, which we use throughout this paper, reads

$$\sigma(f_1(t_i), f_2(t_j)) = \bar{f}_2(t_j) - \bar{f}_1(t_i). \quad (5)$$

Figure 6(a)–(b) illustrates the effect of this conservative comparison function, which readily indicates similarity. This property is beneficial for our overall approach as only the subsequent similarity line filtering (Sec. 4.5) constrains similarity to the desired level. The required local extrema of  $f_1(t)$  and  $f_2(t)$  are precomputed before similarity matrix computation and discarded afterward since the computational cost of the matrix outweighs their recomputation. The same holds for smoothing of  $f_1(t)$  and  $f_2(t)$  prior to the computation of the local extrema.

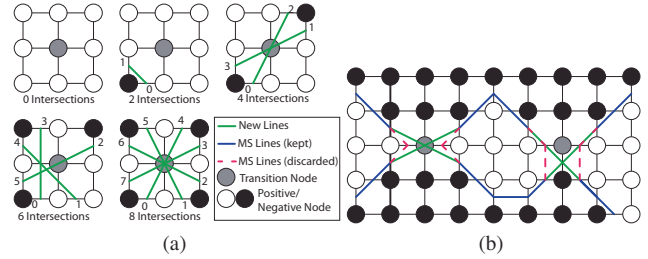


Fig. 5. (a) Exemplary cases for our similarity line extraction scheme around a transition node. (b) Example of our modified marching squares algorithm to extract similarity lines. Both transition nodes in this figure were generated from negative values.

### 4.3 Similarity Matrix Computation

Similarity matrix computation starts with the retrieval of the time-discretized functions  $f_1(t)$  and  $f_2(t)$ . As discussed in Sec. 4.1,  $f_1(t)$  and  $f_2(t)$  can be identical, originating from different points in the same data, or even belong to different data sets. We obtain the time-discretized  $f_1(t)$  and  $f_2(t)$  directly from the time steps of the field data. In a second step, smoothing may be applied to  $f_1(t)$  and  $f_2(t)$  and their local extrema are determined (Sec. 4.2). The value of the similarity matrix element  $\mathcal{S}_{ij}$  is then simply obtained as  $\sigma(f_1(t_i), f_1(t_j))$  and  $\mathcal{C}_{ij}$  as  $\sigma(f_1(t_i), f_2(t_j))$ .

### 4.4 Similarity Line Extraction

Some approaches have been proposed so far for extracting the similarity lines, i.e., the curves of high similarity, from similarity matrices. While many approaches utilize threshold or contour concepts for extracting the lines from nonnegative comparison functions, these approaches are subject to a common issue: since the underlying problem is a valley extraction problem rather than a contour problem, the results typically suffer from disrupted and deviating lines. To the best of our knowledge, ridge (valley) extraction techniques, e.g., according to Eberly [7], have not been used so far for extracting the similarity lines. However, our attempt to extract them using these techniques suffered from disrupted lines too—not due to varying level of the comparison function but due to intersection of the lines. As documented by Eberly, and also due to the fact that codimension-1 ridge extraction [31] is usually achieved based on marching cubes algorithms, ridges obtained this way cannot intersect or branch, in contrast to the similarity lines in the similarity matrices. This motivated our modified marching squares approach together with our *signed* comparison function  $\sigma$  (Eq. 5) for generating the similarity matrices  $\mathcal{S}$  and  $\mathcal{C}$  and extracting the lines therefrom. Running the traditional marching squares algorithm on  $\mathcal{S}$  or  $\mathcal{C}$  at isolevel  $c = 0$  would theoretically produce the desired lines of temporal similarity. However, as similarity lines often intersect, contours as obtained by marching squares tend to form “islands” instead of the desired long intersecting lines, as shown in Fig. 4(a) (not to be confused with the ambiguous cases of traditional marching squares).

In its node-classifying pass, the traditional marching squares algorithm marks each node either as *positive* or *negative*, depending on the sign of the reduced field  $\sigma - c$  at the node (“equal” cases are avoided by perturbation). Our modified version additionally introduces *transition nodes* and provides an additional set of marching squares cases to handle them. A node becomes a transition node when its two horizontal (or vertical) neighbors have sign opposite to that of the node. In a second pass, we apply a connected component labeling considering the 8-neighborhood of each transition node. From each connected component of transition nodes only the transition node is kept whose original value is closest to zero. This guarantees that no other transition nodes exist in the 8-neighborhood of a transition node.

Then, a standard marching squares pass is performed, but skipping all cells that include a transition node. We constrain our description here to our modifications and refer to the literature (e.g., Lorensen and Cline [24]) for the remaining details on traditional marching squares. Finally, we perform a pass with our extended cases considering the 8-

neighborhood of each transition node (Fig. 5(a)). For each of them we look at the intersections of the surrounding cell edges (those that do not connect to the transition node) and generate line segments depending on the total number of intersections  $n$ . The intersection points  $p_i$  with  $0 \leq i < n$  are labeled counterclockwise with increasing local IDs, starting from the bottom left. Then the segments  $(p_i, p_{(i+\frac{n}{2})})$ ,  $\forall i < n/2$  are generated (exemplified in Fig. 5(b)), i.e., they connect “opposite” intersections. The enforcement of crossings is based on a heuristic that proved to work well in our experiments (e.g., Fig. 4(b)). Since the decision of crossings is application-dependent, the researcher can edit the result during the overall visual analysis. One advantage of our approach (especially with respect to the traditional thresholding) is that the extraction of the lines does not require any parameters. Nevertheless, as discussed above, the smoothing and persistence parameter settings allow one to suppress noise and choose the right scale (see Matassini et al. [29] for a detailed discussion of the general topic of noise reduction in recurrence plots).

#### 4.5 Similarity Line Filtering

The extraction of similarity lines as described in Sec. 4.4 is rather conservative. Since similarity is highly application-dependent, a large variety of filtering approaches can be applied to these lines. We found two simple filter criteria particularly useful: line slope and similarity within a time window (Fig. 4(c) and Fig. 4(d), respectively).

First, the line is filtered by its slope (restricting it between *minimum slope*  $s_{\min}$  and *maximum slope*  $s_{\max}$ ). For instance, we may filter for lines with positive slope, additionally discarding close-to horizontal and close-to vertical lines as shown in Figures 4(c) and 6(d). The filter is applied to every vertex of the line and if the test fails, the point is removed, possibly splitting the line in two. Estimation of the slope involves another parameter: the euclidean distance (*slope scale*  $\hat{s}$ ) between the two samples on the line which are used to measure the slope. A large  $\hat{s}$  makes the filter less sensitive to small-scale noise, but may also lose small-scale detail.

Subsequently, we filter each point of the line on the basis of the similarity within a sliding time window (*window similarity size*  $\hat{r}$ ) centered at the point. Note that for computing the similarity matrix, the signal was normalized during comparison, with respect to two surrounding local extrema (Sec. 4.2), typically producing many (parts of) similarity lines that do not represent sufficient similarity from an application point of view. These are rejected in this step. Every line segment defines a time window (Sec. 3.1), i.e., an axis-aligned bounding box in the matrix. Within this window the original signals  $f_1$  and  $f_2$  are normalized and compared to each other, using the root mean square error metric (RMS). The considered point is deleted if the resulting error exceeds a user-defined threshold (*window similarity RMS*  $r$ ). The RMS comparison makes the overall similarity detection more restrictive with increasing  $\hat{r}$  and decreasing  $r$ , making it more specific (Fig. 6(b)–(c) and 4(d)). Note that if no respective window exists because the line is too short, the line is discarded. Other methods for filtering can be easily integrated in our approach.

### 5 VISUALIZATION OF TEMPORAL SIMILARITY

The similarity lines obtained in Sec. 4 can be used for temporal similarity visualization in manifold ways.

#### 5.1 Interactive Inspection of Signals

The simplest visualization mode is to pick one or two points in the spatial view and to inspect the respective signals as well as the corresponding self-similarity or cross-similarity matrix together with the resulting similarity lines. This approach can serve for inspection of the temporal similarity at a point, the temporal relation between locations that are of interest beforehand, and for validation and inspection in the context of the advanced techniques described below. It also allows for parametrization of the different filter criteria.

#### 5.2 Similarity Line Replay

The technique described in Sec. 5.1 provides an overview of temporal similarity for the signal(s) at the (two) selected point(s), in which

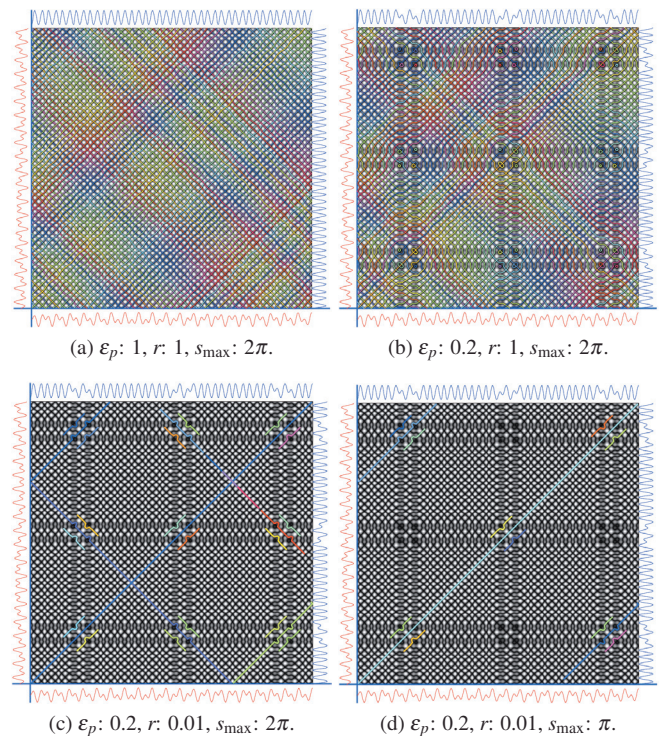


Fig. 6. Similarity lines with different signal comparison and filter settings ( $s_{\min}$  is always 0). The underlying signal is  $\sin(x) \cdot \sin(\pi x)$ . (a) Unfiltered result. (b) Restriction to “double mountain” patterns by means of  $\epsilon_p$ , but other detected similarities persist. (c) Sliding window RMS threshold  $r$  constrains detection to “double mountain” pattern. (d) Slope thresholds  $s_{\min}$  and  $s_{\max}$  restrict to positive time correlation.

each similarity, or correlation, is present as a similarity line. As a dual approach to this mode, the user can pick one of the similarity lines and concurrently replay the (two) time-dependent field(s) side by side, with the selected points marked. The used replay speeds are kept proportional to the slope of the similarity line, rendering the process at equal speed in both views.

This mode provides both temporal and spatial context, and is therefore particularly useful for inspection and reasoning. Another powerful application are cases in which the similarity line exhibits very high or very low slope, i.e., there is a large time scale discrepancy in similarity of the two signals. While visual detection of these similarities would be impossible with realtime replay of the data, our similarity line based replay adjusts the speeds of the replay such that the phenomena appear synchronously.

#### 5.3 Automatic Detection of Self-Similarity

The building blocks described so far required spatial user input: the selection of one or two points. While this can represent a sufficient technique if locations of interest are known a priori, this pointwise technique is inefficient for spatial exploration of large unsteady fields.

Since the extracted similarity lines provide a measure for temporal similarity, we can utilize them for automatic detection of self-similar processes. As this approach operates on data points of the field independently, it is based on  $\mathcal{S}$ .

This mode addresses the research question if a given point in the field is part of a self-similar temporal variation at a given *instant* in time. To this end we construct for each point in the spatial domain the respective self-similarity matrix, extract all filtered similarity lines and mark the point in all time steps where time intersects one of these similarity lines. As demonstrated in Sec. 6.3 this technique is not only able to reveal periodic processes such as von Kármán vortex streets—it also represents a basis for a more detailed analysis based on spatio-temporal similarity clustering described in Sec. 5.4. To provide tem-

poral context, the data is displayed in the space-time domain, i.e., with depth representing time (see, e.g, Fig. 9(b)).

#### 5.4 Spatial Clustering of Temporal Similarity

The basic idea of this step is to aggregate similar voxels not only temporally but also spatially and to analyze cross-relations between these space-time aggregates. We do this by computing a self-similarity matrix for every cell of a data set, as well as two cross-similarity matrices, one with its bottom and one with its right neighbor (throughout this paper we address only hexahedral structured grids, but the extension to unstructured grids is straight-forward). We then extract similarity lines in the cross-similarity matrices, imposing a  $s_{\min}$  and  $s_{\max}$  on the lines to assure a line slope sufficiently close to  $\pi/2$ , i.e., we require temporal similarity at similar time scale between neighboring pixels.

In the space-time volume, the self-similarity matrix and the two cross-similarity matrices are used to measure similarity to temporal and spatial neighbors, respectively. We establish similarity clusters based on the 6-connectivity of each space-time cell (26-neighborhood would induce a much higher computational cost). For the cross-similarity matrices, we quantify similarity by counting the number of similarity lines that intersect the time coordinate of the respective space-time cell. In the self-similarity matrix, we count the number of lines that intersect with that time step as well as the next time step, to establish temporal connectivity. These similarity counts are computed in a pre-processing step in our implementation.

In the subsequent aggregation step, we apply connected component labeling by a region growing procedure, iteratively considering each space-time voxel as a seed for a new region that has not yet been assigned to a region. We grow the region from one space-time voxel to the other if the respective line count (i.e., the number of detected similar processes) is above a user-defined threshold. This can be accomplished at almost interactive rates, depending on the size of the data set. Note that the shape of a resulting cluster is independent of the chosen seed, and that continuously varying signals will result in a common cluster, which is a desired property in many applications. Likewise, moving regions of temporal similarity will get recognized as single space-time structures. Finally, clusters with a size below a user-defined threshold can be removed to reduce visual clutter.

Our clustering is comparably robust against noise as it considers a large number of similarity matrices and lines (a gap in the lines only leads to small “holes”). We investigate the impact of noise in Sec. 6.1. In the case of substantial noise, choosing a lower similarity connectivity threshold for more robust detection, however, at the cost of less clearly defined clusters, may be a worthy trade-off.

For visualization, the space-time volume containing the clusters is uploaded to the GPU to a 3D texture, every voxel containing its cluster ID. A color is assigned to each cluster and the cluster volume is rendered alongside with the spatio-temporal data set (in gray level for context) using raycasting (e.g, Fig. 9(b)). The user can interactively browse through time which moves the volume(s) w.r.t. the clipping plane in front of the viewer. Cells directly behind the clipping plane (i.e., at the current time) are rendered opaque with accentuated contours to better convey the cluster extent at the chosen time (Fig. 1(a)).

#### 5.5 Cluster Comparison

The relation between the clusters can provide valuable insight. To this end, we determine the pairwise similarity of all cluster pairs on-the-fly and visualize it via the width of cluster-connecting lines (Fig. 1(a)). The similarity of two clusters is measured by means of their masters. The master of a cluster is simply the space-time cell that it most strongly connected to its six neighbors (in terms of similarity line intersection (Sec. 5.4)). The cross-similarity matrix between pairs of masters, i.e. of the signals at the masters, is then computed and the length of the longest similarity line therein represents the cross-cluster similarity. Other metrics such as the line count or total length (this would however not be invariant to the frequency of temporal variation) can easily be included in our approach. Using the clusters masters for cross-comparison of clusters (instead of comparing single spatio-temporal cells) drastically reduces the number of cross-similarity eval-

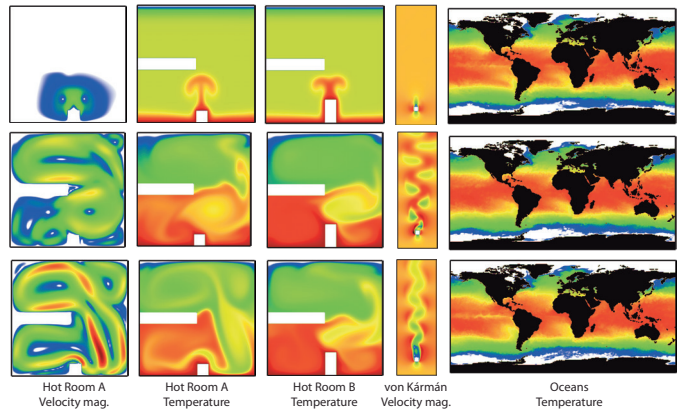


Fig. 7. Overview of data sets. Selected time steps with time progressing from top to bottom.

uations from the square of all pixels, to the square of all clusters. One can reduce this even further by only considering clusters meeting certain criteria, e.g., their size. This is important as without this reduction the cross-similarity tests would be infeasible and the resulting connections would lead to significant clutter when displayed. Basing the cross-cluster similarity measurement only on a line of one similarity matrix can make it susceptible to noise (although we did not experience this issue in our experiments). We suggest that for scenarios featuring strong noise the width of the cluster-connecting lines could be based not only on the master of the cluster but on multiple cells of a cluster.

Cluster-connecting lines can be selected (green line in Fig. 1(a)) for a detailed investigation of the similarity relation by means of the cross-similarity matrix (Fig. 1(b)). Amongst others this also shows why the two clusters were classified the way they were, providing the possibility to adjust the filter criteria, thus triggering the reevaluation of all cross-connections between clusters.

## 6 RESULTS

We implemented our approach using a CUDA-based raycaster for the space-time view, combined with OpenGL geometry to draw the links between clusters as well as the domain outlines. We demonstrate our approach by means of measured and simulated data sets (Fig. 7 and Table 1). Due to the fact that all parts of our approach are easily parallelizable, simple interaction operations like the inspection of signals by selecting single points, or similarity line replay, run interactively without any precomputation in our tests. This means that the signal comparison and the line filtering can be interactively adjusted with interactive feedback. We performed our evaluation on a Core i7 with 2.66 GHz using OpenMP, 8 GB of main memory, and an NVIDIA GTX580 with 3 GB of memory. The space-time visualization part of our method basically boils down to volume raycasting and runs at interactive rates. However, extracting the similarity information required for the detection of self-similarity and clustering is expensive, as it requires per spatial point the computation of similarity lines, and additionally two cross-similarity line sets in the case of clustering. Yet, these computations can be carried out independently from each other and distributed across a cluster (timings in Table 1). Our code was not explicitly tuned and there should be potential for large speedups.

Table 1. Data sets for evaluation. Timings list the duration of the connectivity extraction phase (Sec. 5.4) for our results on an 8 Node cluster. Timings may vary with different parameter settings.

Name	Width × Height	Timesteps	Timing
von Kármán Vel. Mag.	301 × 101	800	4 Mins
Hot Room A Temp.	101 × 101	1600	8 Mins
Hot Room A Vel. Mag.	101 × 101	1600	7 Mins
Hot Room B Temp.	101 × 101	1600	8 Mins
Ocean Temp.	360 × 180	1826	89 Mins

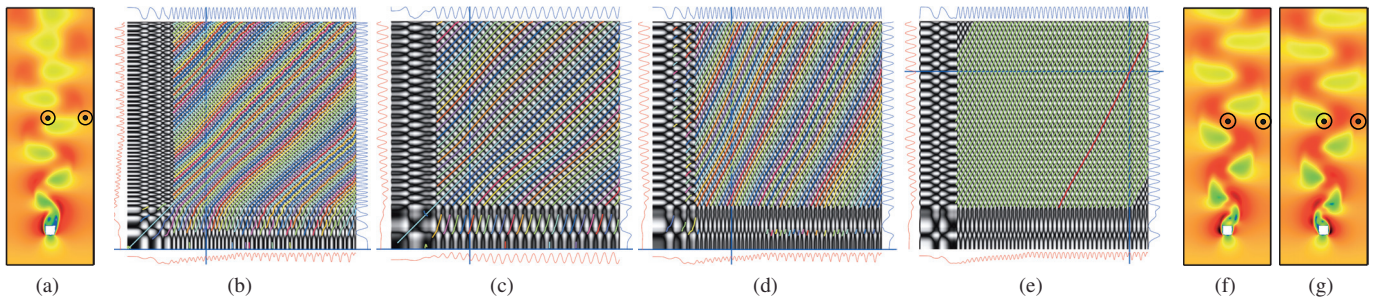


Fig. 8. Interactive inspection. Single points or pairs are selected in the data set (a). For these, self-similarity ((b), (c)) as well as cross-similarity lines (d) are visualized. The similarity line replay mode follows a selected similarity line (red in (e)) through time, playing the time-dependent data set in two windows (f) (abscissa), (g) (ordinate) at respective speeds.

We provide an overview on all presented visualization techniques as well as the impact of noise by means of the von Kármán CFD data set (Sec. 6.1). Then we evaluate the similarity between different buoyant CFD simulations (Sec. 6.2). Finally, we look for periodic processes at different scales in measured ocean data (Sec. 6.3).

### 6.1 Overview with von Kármán Data Set

The data set used in this section is a 2D time-dependent CFD simulation of a von Kármán vortex street (Fig. 7). We use this example both for providing insight and to exemplify the techniques and methodology described in the previous sections.

The interactive inspection of signals is demonstrated on this example in Fig. 8. When points are picked in a data set directly, the respective similarity matrix and similarity lines are computed and visualized on the fly. In this example, we selected one point in the center of the data set and one right to it (Fig. 8(a)). The similarity lines of the left and the right point (Figure 8(b) and (c)) show that there is a large number of recurrent processes running at comparable frequencies with respect to each other (45 degrees slope of similarity lines). They also show that recurrent behavior does not start until a certain point in time. Their (varying) period can be judged from the distance between the lines. The cross-similarity lines computed from both points (Fig. 8(d)) show that the variation of the two compared signals does not take place at the same frequency but that the frequency at the left point is about twice the one at the right.

While looking at the similarity lines already provides some insight regarding temporal variation of the data set, tracking the lines and showing the meaning of a similarity line is crucial in real data. To this end, we selected a similarity line (Fig. 8(e)) for similarity line replay. As each point in the similarity line defines two time instances, two views are employed to display them side-by-side (Fig. 8(f) and 8(g)). Replay in Fig. 8(g) is about twice the speed of Fig. 8(f) but the variation in the replay is perceived consistent.

The manual selection of points is very helpful when the user already has an idea what he is looking for, but can be tedious in the case of more complex data sets, or absence of a priori knowledge. To overcome this issue, we can run a self-similarity analysis for every spatial point in the data set (Sec. 5.3) and visualize the resulting space-time volume (Fig. 9(a)). This shows at which spatial point at which times recurrent processes exist. Separating the structures using clustering (Sec. 5.4) helps the visualization and enables further analysis techniques. The coloring shows that the von Kármán vortex street consists of three large clusters (Fig. 9(b)). One cluster consists of clockwise vortices, the other of counter-clockwise vortices, and the smallest cluster captures the area in between that exhibits temporal change at double frequency. The clusters are also compared amongst each other (Sec. 5.5). By filtering the similarity lines, the user can define the kind of similarity appropriate for his needs. In this case, we restricted the angle to about 45 degrees. Hence, there is a thick link between the left and the right cluster, representing strong temporal similarity because their processes are basically mirrored horizontally, and two thin links to the middle cluster because its signals differ and run at higher frequency. This shows in a fully spatio-temporal manner what has al-

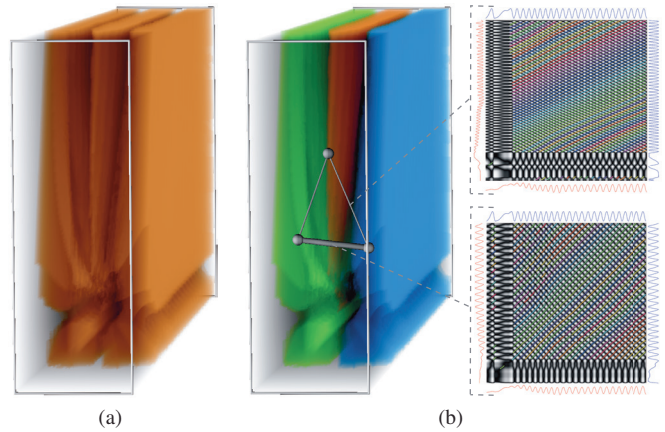


Fig. 9. Visualizing self-similarity (a) and clusters (b) in the von Kármán data set. Thick lines connect regions with similar frequency (defined by  $s_{\min}$  and  $s_{\max}$ ), while thin lines connect regions that are not as closely related (the lines in the top similarity matrix are filtered in this example and shown with relaxed angle constraints for demonstration purposes).

ready been detected for single spatial points earlier (Fig. 8).

We exemplify the robustness of our technique by superimposing the data with increasing levels of noise (Fig. 10). Compared to the reference with  $n = 0$  (Fig. 10(a)),  $n = 0.1$  (Fig. 10(b)) shows no significant difference. With stronger noise ( $n = 0.2$ , Fig. 10(c)) gaps in formerly straight similarity lines occur, which possibly results in the splitting of clusters. With even stronger noise ( $n = 0.3$ , Fig. 10(d)), disruptions of similarity lines occur more often, leading to the splitting of clusters in thin cluster regions. The results show that small to moderate noise has only minor impact, while even with strong noise basic characteristics are preserved.

### 6.2 Inter-Data Similarity with Hot Room Data Set

This data set results from a time-dependent 2D simulation of air flow within a closed container, driven by buoyant forces imposed by a heated bottom plate and a cooled top plate. To provoke transient aperiodic flow, the container exhibits two barriers. Two variants of this data set are used. They are part of a design study with different configuration and only differ in the size of the barrier at the bottom wall. Variant “Hot Room A” exhibits a square obstacle there, whereas “Hot Room B” features a quadrangular obstacle of double height (Fig. 7). Both data sets include velocity and temperature. The data sets (Table 1) consist of 1600 timesteps, but we skip the first 800 in our analysis to focus on the interesting effects between time steps 800 and 1600.

We compare the temperature of Hot Room A both to the velocity magnitude of Hot Room A as well as the temperature of Hot Room B (Fig. 11). First, we compare the temperature fields of the two simulations. The difference of the shape of the obstacle at the bottom wall results in a significant change in the time frame of recurrent processes

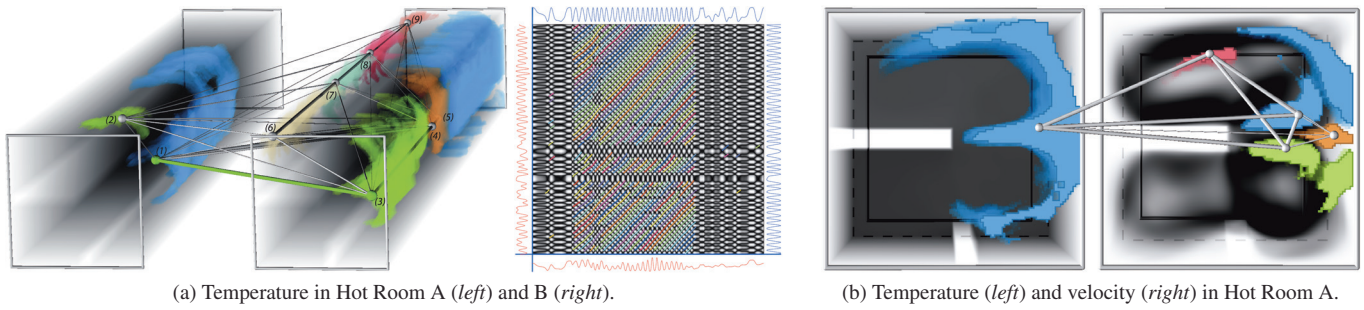


Fig. 11. Cross-comparisons between (a) temperature development in Hot Room A and Hot Room B as well as (b) temperature with velocity magnitude of Hot Room A. In (a), the onset of the von Kármán Vortex Street is much later in A than in B. On the other hand, B exhibits two interruptions of temporal similarity, resulting in an additional cluster (orange). Both the time frame of the cluster in A as well as the two interruptions in B appear in the similarity matrix of the respective masters. In (b), the cluster connecting lines show that two major and one minor clusters of the velocity magnitude are similar to the major temperature cluster, while one velocity cluster (orange) substantially differs.

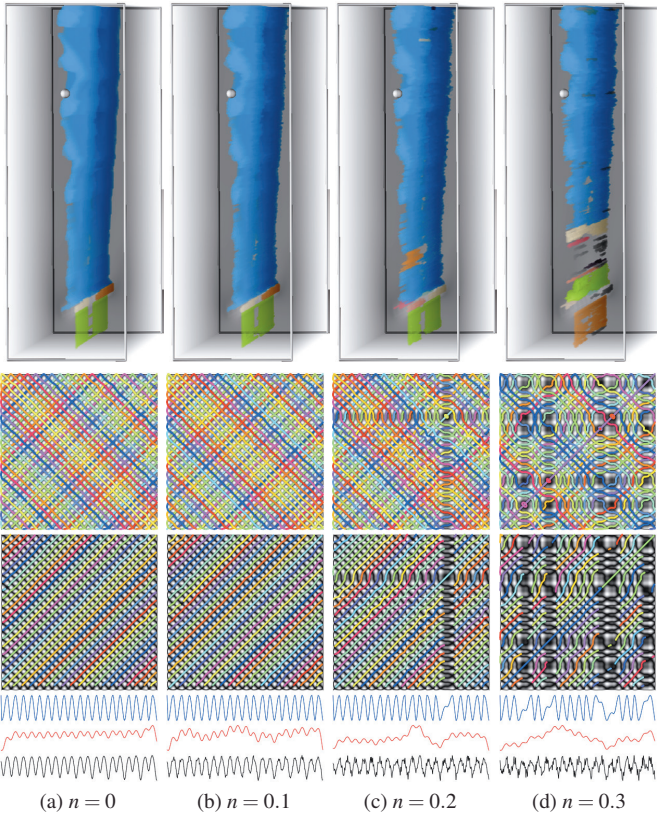


Fig. 10. Time subset (500 to 800) of the von Kármán data set with the original signals  $s^i$  being superimposed with noise:  $s_n^i(t) = s^i(t) + n \cdot \alpha \cdot u^i(t)$  with  $u^i(t)$  delivering uniform noise in  $[-1, 1]$  and  $\alpha$  being the data range of  $s^i$ . Parameter settings from Fig. 9 were adjusted to a lower cluster size threshold and only central clusters are visualized to clearly show the effects of noise. Similarity matrices (both unfiltered and filtered) are shown for the spatial position of the gray sphere.

as visualized in Fig. 11(a). Most prominently, data set A features only two clusters of sufficient size, compared to seven in B. It can also be seen that the recurrent processes in B start earlier (front-most clusters (1) and (3) as well as their similarity lines (Fig. 11(a) right)). The cross-similarity lines show that the processes run at approximately the same frequency. The temporal disruption of clusters (3), (4) and (5) as well as the temporal offsets of the clusters also show up in the cross-similarity lines (note that the masters of clusters (3), (4) and (5) are virtually identical). The thickness of the cluster-connecting lines shows that there is high similarity between clusters (1) from A and (3), (4), and (5) from B. This means that despite all differences between

the two data sets, the spatial points right of the horizontal obstacles in both data sets exhibit similar variation, however, at different points in time. This is in accordance with the investigation of the velocity fields: both exhibit von Kármán vortex street behavior. Furthermore, in B the clusters (6), (7), (8) and (9) also exhibit a strong connection. Hence, there are two different structures in B, both are periodically interrupted: the von Kármán vortex street and the recurrent behavior in the left upper corner induced by the vortex street. In contrast, (2) in A does not exhibit a strong connection to any other cluster.

As shown with Figures 9(b) and 11(a) (disregarding the similarity matrices), just looking at the similarity clusters and their connections already gives a good overview of the behavior of the data set(s). Additionally browsing through time allows for a more detailed investigation. Finally, we compare the velocity magnitude with temperature of the Hot Room A data set (Fig. 11(b)). At first glance, it becomes apparent that with the very same settings in the generation of both volumes, the velocity magnitude splits into three regions while there is only one large cluster for temperature. The thickness of the connecting lines shows that the behavior of the large temperature cluster is similar to the bottom and top clusters in the velocity data, but not to the middle cluster. This visualizes the fact that the middle cluster exhibits, as in Sec. 6.1, double frequency. The fact that temperature contains only one cluster per von Kármán vortex street visualizes that the vortex street is driven by hot flow from the bottom, hence the double frequency in local motion does not show up in temperature.

### 6.3 Periodicity in Ocean Data Set

The ocean temperature data set provides sea surface temperature measured from satellites and combined with direct in situ measurements, provided by the Group for High-Resolution Sea Surface Temperature. Originally, the data set features a spatial resolution of  $1440 \times 720$  at a time resolution of one day, but we have downsampled the data to  $360 \times 180$  and a time resolution of two days to avoid out-of-core volume rendering. The time frame of our analysis is the complete 1990s.

In our experiment, we detected two major effects interacting with each other: the Intertropical Convergence Zone (ITCZ) and the El Niño-Southern Oscillation. The ITCZ is the area encircling the Earth near the equator where winds originating in the northern and southern hemispheres come together (Fig. 12). It is a key component of the global circulation system. El Niño is a quasiperiodic climate pattern that occurs across the tropical Pacific Ocean roughly every five years. Its effects include warmer ocean temperatures across the central and eastern tropical Pacific Ocean, increased convection or cloudiness in the central tropical Pacific Ocean, and low (negative) values of the SOI (Southern Oscillation Index).

First we applied moderate smoothing to the original data to preserve small-scale features (Fig. 12). It can be seen that the generated clusters classify the ITCZ as a region in which periodic behavior occurs. However, under El Niño conditions, the ITCZ region stops being periodic according to the measurement data. This is reflected by the fact that there are no continuous clusters covering all considered time steps in



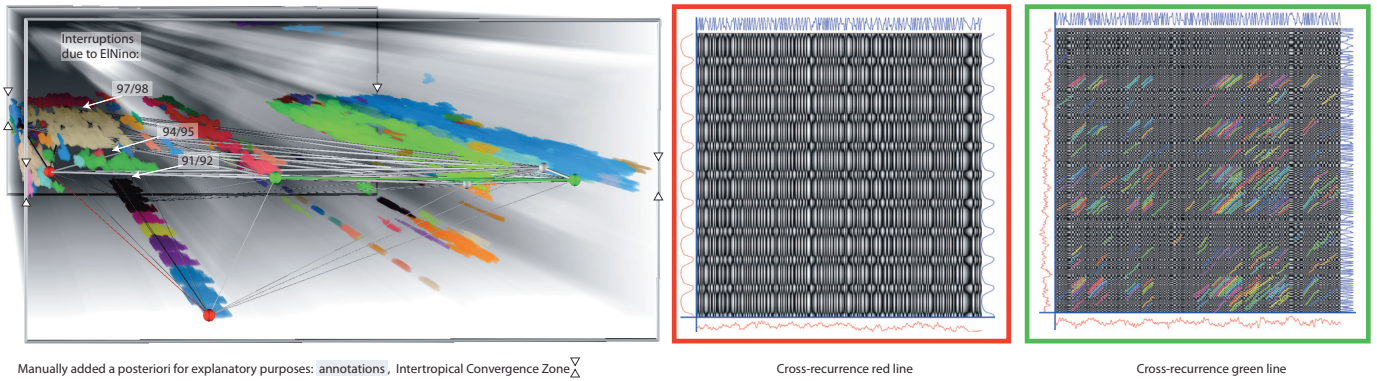
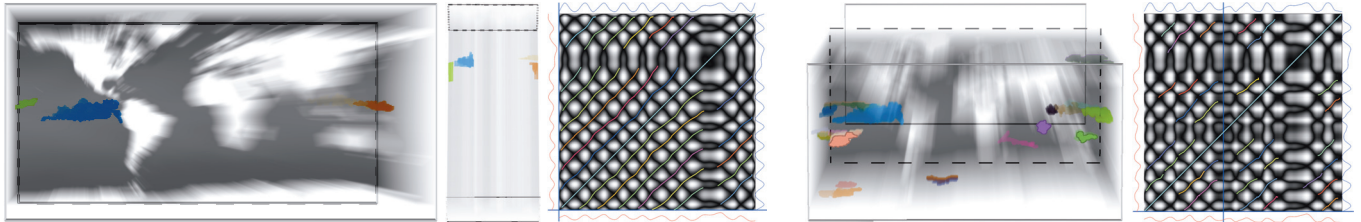


Fig. 12. Similarity clustering with minor signal smoothing only ( $\delta : 4, \epsilon_p : 0.04, s_{\min} : 0.37, s_{\max} : 1.1, \hat{s} : 15, \hat{r} : 100, r : 0.2$ ). Links between the ten largest clusters are shown and cross-similarity matrices for red and green selection. Annotations were added a posteriori for explanatory purposes.



(a) Relaxed settings lead to many long similarity lines and clusters that are more robust to fluctuation in the signal ( $\delta : 17, \epsilon_p : 0.3, s_{\min} : 0.24, s_{\max} : 1.1, \hat{s} : 15, \hat{r} : 100, r : 0.11$ ). Thus only the strongest El Niño from 1997/98 is detected (left: front view, center: top view). (b) Stricter Settings than in (a) ( $\delta : 21, \epsilon_p : 0.1, s_{\min} : 0.24, s_{\max} : 1.1, \hat{s} : 15, \hat{r} : 180, r : 0.2$ ), leading to increased sensitivity, thus additionally detecting the El Niño from 1994/95 (orange).

Fig. 13. Comparably strong smoothing parameters and inverse clustering detects non-periodic behavior in otherwise predominantly periodic signals.

the Eastern Pacific. Instead, the formation of separated clusters correlates to the El Niño event. The cross-similarity plot of two clusters from the ITCZ, at different spatial locations, shows a strong cross-correlation, that is, however, interrupted significantly around 1995 and 1998, two years in which El Niño took place. Because we apply minor smoothing only, we mainly handle small-scale features and accordingly get clusters with small-scale similar behavior. However, if the signal is sufficiently smooth, large-scale similarity still can be detected. This can be observed when looking at the cross similarity lines belonging to the red cross-cluster link in Fig. 12. The signal (the left red graph) shows large-scale fluctuation stemming from the seasons, which applies to virtually all sea points in the data set, but it is detected here because of the exceptional signal smoothness.

To be able to detect similar processes not only on the scale of days but also months, we apply substantial smoothing for a second application (Fig. 13). Clustering similarity like in previous examples, with this setting, produces one large cluster covering almost the complete domain, except for isolated water masses such as the Big Lakes. This is because they all share the common temperature fluctuation linked to seasons. In this case, not similarity or recurrence itself is interesting, but rather whether and when non-periodic phases occur in otherwise periodic signals. We considered a point as being part of a recurrent process if it features at least one time period (in this case the length of 3 years or 500 time steps) with uninterrupted recurrent behavior as indicated by similarity lines. For these points, we cluster with an inverse clustering criterion, i.e., merging spatio-temporal points that would normally not be merged. Like in the previous example with moderate smoothing, this again reveals the El Niño effects. Fig. 13(a) depicts the effect of a particularly strong El Niño (one of the strongest ever recorded) in 1997/98. When slightly decreasing the signal threshold and the smoothing, also the weaker El Niños in 1991 and 1994 are detected (Fig. 13(b)). This means that we detect the El Niño both by means of the disturbance of small and large scale recurrence.

Our findings correlate not only with geospatial literature (e.g., Clarke [5]) but also with the effects described in the concurrent work by Köthur et al. [19], who achieved similar results with sea level instead of temperature data. In contrast to our technique that works with the granularity of one spatial point in time, they cluster complete time

steps. This allows us, in contrast, to extract (and visualize directly) not only when but also where effects occur. It also means that disturbances in a certain spatio-temporal region do not interfere with a cluster covering the same time step in a remote location. Furthermore, clusters and empty regions can be visually inspected to investigate the details behind the clustering. Finally, our technique allows for cross-comparison between different quantities like temperature and sea level.

## 7 CONCLUSION

We presented an interactive technique for visualizing temporal similarity in fields, based on similarity matrices. It allows for visual exploration of both temporal and spatial relationships in terms of temporal variation. The whole pipeline supports visual interaction and inspection, and thus provides a flexible time-dependent data analysis technique. We demonstrated the utility of our approach by means of data both from simulation and measurement. Our method is particularly well-suited for continuous signals and might not perform well for discontinuous signals like movies. Furthermore, the size of the data sets is limited by GPU memory for rendering the spatio-temporal clusters, and for large data sets computing the similarity information required for clustering is an expensive task if no optimization techniques are applied. Nevertheless, our approach lends itself well to parallelization. Our work introduces a new concept which requires further evaluation (e.g., with respect to noise or runtime behavior) and performance optimization (e.g., GPU-accelerated similarity matrix generation) that are beyond the scope of the paper at hand. This particularly includes the comparison with other temporal feature detection approaches as well as a more detailed analysis with a large range of data sets.

For future work, we plan to extend our approach to not only handle similarity lines but also similarity structures. We further intend to apply our method to data along trajectories in vector fields to also account for the Lagrangian frame.

## ACKNOWLEDGMENTS

The authors would like to thank the German Research Foundation (DFG) for supporting the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) and the Collaborative Research Center SFB-TRR 75 at the University of Stuttgart.

## REFERENCES

- [1] J. Ahrens, K. Heitmann, M. Peterson, J. Woodring, S. Williams, P. Fasel, C. Ahrens, C. Hsu, and B. Geveci. Verifying scientific simulations via comparative and quantitative visualization. *IEEE Computer Graphics and Applications*, 30(6):16–28, 2010.
- [2] D. Angus, A. Smith, and J. Wiles. Conceptual recurrence plots: Revealing patterns in human discourse. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):988–997, 2012.
- [3] E. Bautista-Thompson, R. Brito-Guevara, and R. Garza-Dominguez. Recurrencevs: A software tool for analysis of similarity in recurrence plots. In *Proceedings of Electronics, Robotics and Automotive Mechanics Conference 2008*, pages 183–188, 2008.
- [4] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911–920, 2011.
- [5] A. J. Clarke. *An Introduction to the Dynamics of El Nino & the Southern Oscillation*. Academic Press, 2008.
- [6] R. Cutler and L. Davis. Robust periodic motion and motion symmetry detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2000*, volume 2, pages 615–622, 2000.
- [7] D. Eberly. *Ridges in Image and Data Analysis*. Computational Imaging and Vision. Kluwer Academic Publishers, 1996.
- [8] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 454–463, 2000.
- [9] Z. Fang, T. Möller, G. Hamarneh, and A. Celler. Visualization and exploration of time-varying medical image data sets. In *Proceedings of Graphics Interface 2007*, pages 281–288, 2007.
- [10] R. Fuchs, J. Waser, and M. Gröller. Visual human+machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1327–1334, 2009.
- [11] S. Gerber, P. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1271–1280, 2010.
- [12] M. Glatter, J. Huang, S. Ahern, J. Daniel, and A. Lu. Visualizing temporal patterns in large multivariate data using modified globbing. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1467–1474, 2008.
- [13] D. Guo, J. Chen, A. MacEachren, and K. Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474, 2006.
- [14] H. Jänicke, M. Bottinger, U. Mikolajewicz, and G. Scheuermann. Visual exploration of climate variability changes using wavelet analysis. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1375–1382, 2009.
- [15] H. Jänicke and G. Scheuermann. Knowledge assisted visualization: Steady visualization of the dynamics in fluids using  $\epsilon$ -machines. *Computers and Graphics*, 33(5):597–606, 2009.
- [16] J. Kehrer, P. Filzmoser, and H. Hauser. Brushing moments in interactive visual analysis. *Computer Graphics Forum*, 29(3):813–822, 2010.
- [17] J. Kehrer, F. Ladstadter, P. Muigg, H. Doleisch, A. Steiner, and H. Hauser. Hypothesis generation in climate research with interactive visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1579–1586, 2008.
- [18] E. Kononov. Visual recurrence analysis software. <http://nonlinear.110mb.com/vra/>.
- [19] P. Köthur, M. Sips, J. Kuhlmann, and D. Dransch. Visualization of geospatial time series from environmental modeling output. In *Short Paper Proceedings of the 14th Eurographics/IEEE VGTC Symposium on Visualization*, pages 115–119, 2012.
- [20] P. Kristensson, N. Dahlback, D. Anundi, M. Bjornstad, H. Gillberg, J. Haraldsson, I. Martensson, M. Nordvall, and J. Stahl. An evaluation of space time cube representation of spatiotemporal patterns. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):696–702, 2009.
- [21] O. D. Lampe and H. Hauser. Curve density estimates. *Computer Graphics Forum*, 30(3):633–642, 2011.
- [22] T.-Y. Lee and H.-W. Shen. Visualization and exploration of temporal trend relationships in multivariate time-varying data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1359–1366, 2009.
- [23] T.-Y. Lee and H.-W. Shen. Visualizing time-varying features with tab-based distance fields. In *Proceedings of the 2009 IEEE Pacific Visualization Symposium*, pages 1–8, 2009.
- [24] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.
- [25] K.-L. Ma. Machine learning to boost the next generation of visualization technology. *IEEE Computer Graphics and Applications*, 27(5):6–9, 2007.
- [26] M. M. Malik, C. Heinzl, and M. E. Gröller. Comparative visualization for parameter studies of dataset series. *IEEE Transaction on Visualization and Computer Graphics*, 16(5):829–840, 2010.
- [27] N. Marwan, M. Carmenromano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5-6):237–329, 2007.
- [28] N. Marwan, J. Kurths, and P. Saperin. Generalised recurrence plot analysis for spatial data. *Phys. Lett. A* 360 (45), pages 545–551, 2007.
- [29] L. Matassini, H. Kantz, J. Holyst, and R. Hegger. Optimizing of recurrence plots for noise reduction. *Phys. Rev. E*, 65(2):021102, 2002.
- [30] R. Peikert and F. Sadlo. Visualization methods for vortex rings and vortex breakdown bubbles. In *Proceedings of the 9th Eurographics/IEEE VGTC Symposium on Visualization*, pages 211–218, 2007.
- [31] R. Peikert and F. Sadlo. Height ridge computation and filtering for visualization. In *Proceedings of the 2008 IEEE Pacific Visualization Symposium*, pages 119–126, 2008.
- [32] T. Pham, R. Hess, C. Ju, E. Zhang, and R. Metoyer. Visualization of diversity in large multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1053–1062, 2010.
- [33] J. Reininghaus, N. Kotava, D. Gunther, J. Kasten, H. Hagen, and I. Hotz. A scale space based persistence measure for critical points in 2d scalar fields. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2045–2052, 2011.
- [34] T. M. Rhyne, A. MacEachren, and T.-M. Rhyne. Visualizing geospatial data. In *Proceedings of ACM SIGGRAPH 2004 Course Notes*, 2004.
- [35] F. Sadlo, R. Peikert, and M. Sick. Visualization tools for vorticity transport analysis in incompressible flow. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):949–956, 2006.
- [36] T. Salzbrunn, C. Garth, G. Scheuermann, and J. Meyer. Pathline predicates and unsteady flow structures. *Visual Computer*, 24(12):1039–1051, 2008.
- [37] A. Sanderson, G. Chen, X. Tricoche, D. Pugmire, S. Kruger, and J. Breslau. Analysis of recurrent patterns in toroidal magnetic fields. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1431–1440, 2010.
- [38] 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.
- [39] D. Vasconcelos, S. Lopes, R. Viana, and J. Kurths. Spatial recurrence plots. *Phys. Rev. E* 73, 2006.
- [40] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.
- [41] J. Waser, R. Fuchs, H. Ribic andic and, B. Schindler, G. Blöschl, and E. Gröller. World lines. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1458–1467, 2010.
- [42] J. P. Zbilut and C. L. Webber. Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A*, 171(34):199–203, 1992.