Time-hierarchical Clustering and Visualization of Weather Forecast Ensembles

Florian Ferstl, Mathias Kanzler, Marc Rautenhaus, and Rüdiger Westermann

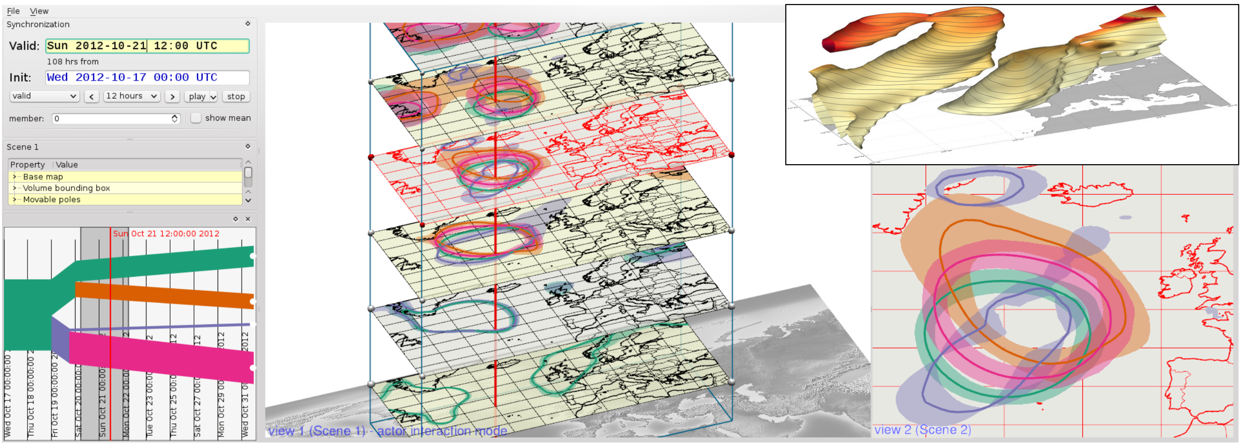
Valid: Sun 2012-10-21 12:00 UTC
108 hrs from
Init: Wed 2012-10-17 00:00 UTC
member: 0

Fig. 1. Interactive exploration of time-varying iso-contours in a weather forecast ensemble: Based on our time-hierarchical clustering (bottom left) the user can animate through visualizations of selected time steps (bottom right). An overview of all time steps is shown as a stacked plot (center), and a representative space-time surface can be shown for a selected cluster (inset, top right).

Abstract—We propose a new approach for analyzing the temporal growth of the uncertainty in ensembles of weather forecasts which are started from perturbed but similar initial conditions. As an alternative to traditional approaches in meteorology, which use juxtaposition and animation of spaghetti plots of iso-contours, we make use of contour clustering and provide means to encode forecast dynamics and spread in one single visualization. Based on a given ensemble clustering in a specified time window, we merge clusters in time-reversed order to indicate when and where forecast trajectories start to diverge. We present and compare different visualizations of the resulting time-hierarchical grouping, including space-time surfaces built by connecting cluster representatives over time, and stacked contour variability plots. We demonstrate the effectiveness of our visual encodings with forecast examples of the European Centre for Medium-Range Weather Forecasts, which convey the evolution of specific features in the data as well as the temporally increasing spatial variability.

Index Terms—Ensemble visualization, uncertainty visualization, meteorological visualization, iso-contours, time-varying data, clustering.

1 Introduction

Ensemble weather forecasts have been well established in meteorology as a tool to provide estimates of the uncertainty inherent in numerical weather predictions [4]. Based on perturbed initial conditions or different forecast model formulations, ensemble methods provide a representative sample of possible future states of the atmosphere [26]. Analysis of the temporal evolution and variability of an ensemble forecast is an important task. For example, a forecaster needs to judge whether a prediction can be trusted in a particular region and forecast time window. Also, information on possible future atmospheric scenarios the ensemble predicts and their likelihood are of interest [9]. Furthermore, spatio-temporal analysis of an ensemble forecast can provide information on where and when ensemble members start to diverge, allowing inference on atmospheric phenomena that cause members to evolve in different ways.

Spaghetti plots have been established as a common visualization tool to analyze the variability of iso-contours in meteorological scalar field ensembles (e.g., [50]). Typically, these plots simultaneously show contours of all ensemble members at a single time-step. For analysis of the temporal evolution and variability of the displayed scalar fields, animation of sequences of spaghetti plots is frequently used (see, e.g., the website of the National Oceanic and Atmospheric Administration's Storm Prediction Center [29]). When viewing such animations, however, connections between contours in consecutive time-steps are difficult to establish, due to the huge amount of visual information that needs to be memorized by the viewer. As pointed out by Gleicher et al. [14, p. 294], animation "requires the use of the viewer's memory and attention shifts to make connections between objects", making it problematic in general for spatio-temporal analysis. A number of visual abstractions of spaghetti plots have been proposed in recent years [42, 48, 11]. They, however, also represent single time-steps and rely on animation to visualize the temporal development.

In this work we develop an alternative approach to analyze the spatio-temporal evolution of iso-contours obtained from ensemble weather predictions. We target a setting in which the ensemble members are started from perturbed but (relatively) similar initial conditions. This is true, e.g., for the operational ensemble forecast produced by the European Centre for Medium-Range Weather Forecasts

- *F. Ferstl, M. Kanzler, M. Rautenhaus and R. Westermann are with the Technical University of Munich. E-mail: ferstlf@in.tum.de, mathias.kanzler|marc.rautenhaus|westermann@tum.de.*

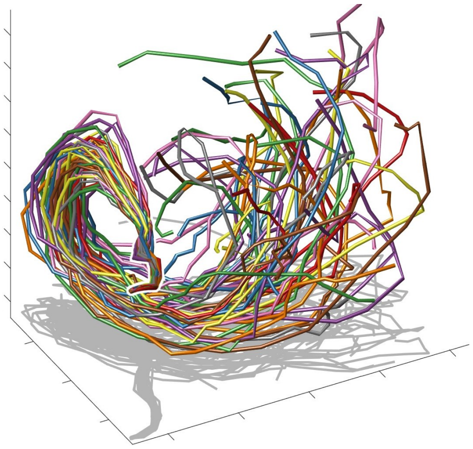

Fig. 2. Multi-run plot [12] of an ensemble of 51 iso-contours to a fixed iso-value in a geopotential height field, over the first 8 days of a weather forecast in time-steps of 6 hours. Each line represents the evolution trajectory of one ensemble member. Each line vertex corresponds to an iso-contour in a specific time-step. The data is projected to a 3D space using a multi-dimensional scaling projection, so that large vertex-to-vertex distances correspond to large dissimilarities between the corresponding contours.

(ECMWF) Ensemble Prediction System (ENS; e.g., [26]). Due to the chaotic nature of the atmosphere and the techniques used to select initial ensemble members, the variability, or spread, of such forecast ensembles on average increases over time [26]. As demonstrated in Fig. 2, this diverging nature can be visualized with "multi-run plots" as proposed by Fofonov et al. [12]. They compute a similarity matrix of all contours and visualize their trajectories using a multidimensional scaling projection. Note, however, that besides restricting the visualization to the three most significant axis in the multidimensional feature domain, the method cannot show where in the spatial domain the spread is large or low.

To analyze the spatio-temporal evolution of such ensemble predictions, we target the following objectives:

- Temporal evolution similarity: Group ensemble members with similar temporal evolution in a specified region and time window (following operational but static practice at ECMWF [9], we assume that the user is interested in grouping ensemble members according to similar forecast scenarios in a user-specified region and time windows, and in analyzing the temporal development of the obtained groups prior to and after the selected window).

- Time-hierarchical variability: Develop approaches that show the hierarchical temporal evolution of such groups of members and that help to determine the spatial and temporal locations at which differences between the ensemble members start to occur.

- Space-time visualization: Develop visual encodings to analyze the temporal evolution of spatial variability in an effective way.

Contribution: To achieve the aforementioned goals, we propose new approaches for clustering weather forecast ensembles with respect to their temporal evolution, and for visualizing the resulting cluster hierarchies in a spatio-temporal context. Our specific contributions are with respect to the following aspects:

- Clustering of iso-contours with similar time evolution: We consider sets of sequences of iso-contours and cluster these sets to find iso-contours with similar dynamics. Each sequence is comprised of iso-contours in a given ensemble member over an arbitrary given time window.

- Time-hierarchical clustering of iso-contours: By taking into account the characteristic behavior of weather forecast ensembles, we propose merging of clusters from a specified time window in the forecast range in time-reversed order. This creates a cluster tree, with the root node being the initial conditions from which the temporal integration of the ensemble simulations has started.

- Space-time cluster visualization: We develop visualization techniques to assess in a single view where variations in the predicted atmospheric states occur and when they start occurring.

We believe that our findings can significantly help to improve and accelerate the analysis of ensembles of iso-contours extracted from meteorological simulation output starting from perturbed but similar initial conditions. In particular, we demonstrate that our approach has significant advantages over the animation of spaghetti plots, since it can reveal information which is difficult to grasp from such plots. We demonstrate the effectiveness of our approach with real-world forecast examples obtained from ECMWF.

2 Related Work

Uncertainty visualization is one of the top challenges in scientific visualization [5, 37]. Uncertainty in scientific data is often estimated by means of ensembles—a representative sample of possible realizations of a simulated phenomenon, obtained by running simulations with different initial conditions and physical models. Such data is typically spatiotemporal, multivariate, and multivalued [22, 28], making the analysis and visualization processes difficult. Typical methods to simplify the data evaluate summary statistics and visualize these using color maps, contours, surface deformation, opacity, boxplots, or glyphs [28, 33, 36].

For vector fields, Wittenbrink et al. [51] propose uncertainty vector glyphs to show the magnitude and angular uncertainty. For time-varying uncertain vector fields, Hlawatsch et al. [18] introduce flow radar glyphs. To consider the transport uncertainty, Otto et al. [32] use particle density functions to obtain an uncertain topological segmentation for Gaussian-distributed steady vector fields and Hummel et al. [19] compare the material transport in time-varying flow ensembles by computing individual and joint vector field variances.

Other visualization techniques for dealing with the complexity of ensemble data include coordinated multiple views, which are commonly used to study multivariate relations via linking and brushing [22] and parallel coordinates [17]. Nocke et al. [30] use coordinated multiple views for climate ensembles. Piringer et al. [34] propose a design on three levels of details for a comparative visual analysis of 2D function ensembles.

Standard approaches for dealing with large and complex data are to restrict to subsets of the data using, e.g., feature based querying [23], or to use clustering [21]. The clustering of iso-contours in 2D is related to the clustering of parametrized curves. This can be achieved using a variety of different geometric distance measures, e.g., as compared by Zhang et al. [52] for 2D trajectories in surveillance videos, or by Oeltze et al. [31] for 3D streamlines of blood flow simulations. In contrast to clustering curves, the clustering of iso-contours based on geometric representations is more involved because iso-contours do not have a natural parametrization and are often composed of multiple, fragmented parts. Several alternative representations have been used to analyze iso-contours like, e.g., rotation and scale invariant features [46] or histograms and iso-surface statistics [7]. In particular, signed distance functions are a popular choice for representing iso-contours because each contour can be directly mapped to a point in a high-dimensional Euclidean space by sampling the corresponding signed distance function on a fixed grid. Bruckner and Möller [6] use signed distance functions to analyze different iso-contours of the same scalar field, Rathi et al. [39] use such functions for shape analysis, and Ferstl et al. [11] have recently proposed to use signed distance fields for clustering ensembles of iso-contours.

The clustering of time series data has been shown to be effective in several domains like computer vision, data mining and visualization. Previous approaches have focused mainly on the clustering of uni- or multivariate time series data, i.e., sequences of scalar values

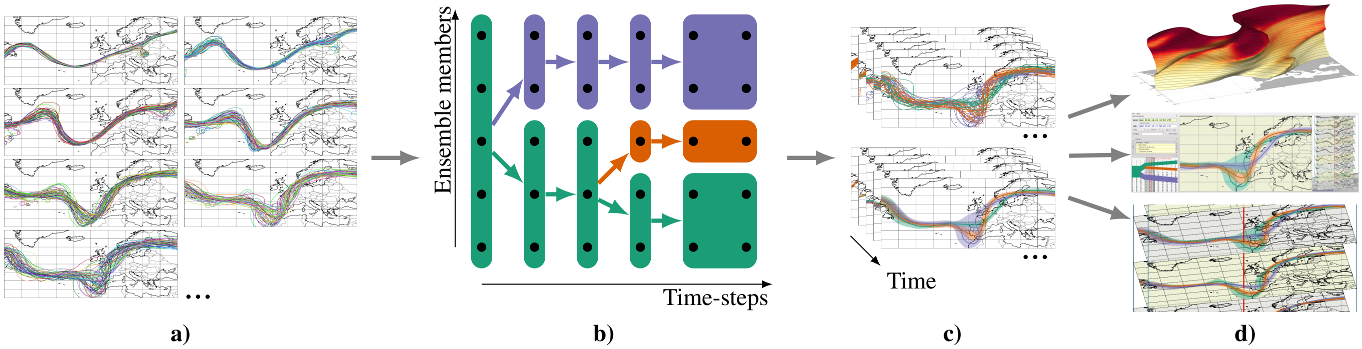

Fig. 3. Method overview: **a)** Ensemble of time-varying iso-contours. **b)** Time-hierarchical clustering of the ensemble members. **c)** Spaghetti plots of clustered contours & variability plots. **d)** Our visualizations: Space-time cluster surface, interactive cluster selection and visualization, stacked time-cuts.

which can be interpreted as vectors and clustered using, for instance, Euclidean distances [35] or root-mean-square distances [49]. Alternative approaches perform clustering based on derived features instead of the raw data [15]. For a thorough overview of approaches for the clustering of time series let us refer to the articles by Denton [8] and Liao [27]. While several works consider the clustering of time-series of data points which are more complex than simple scalars or vectors, to the best of our knowledge, no method yet has considered time series data exhibiting the slowly diverging nature like weather forecast ensembles, which gives rise to the time-hierarchical clustering we propose in this work.

The visualization of variability in ensembles of 2D scalar fields is often performed using spaghetti plots of iso-contours, especially when applied to weather forecast data [50, 38]. Because the resulting visualizations often suffer from visual clutter, several simplifications and visual abstractions have been introduced in recent years. Sanyal et al. [42] use glyphs and graduated ribbons to convey the uncertainty along iso-contours. Different kinds of confidence bands—regions representing the Euclidean spread of a set of iso-contours—are introduced by Whitaker et al. [48], which build upon the concept of statistical band depth, and by Ferstl et al. [11], which build upon the concept of standard deviation of signed distance functions.

The visualization techniques for time-varying 2D scalar ensembles we propose in this work build upon the concept of space-time cubes. Space-time cubes define a three-dimensional coordinate system, where two dimensions refer to space and the third dimension refers to time [16, 24]. Many variants of space-time cubes have been proposed, which we cannot attempt to review here, however, a thorough overview of space-time cubes and their application-specific variations is given in the state-of-the-art report by Bach and co-workers [3]. Andrienko et al. [2] use space-time cubes for the visualization of temporal events at specific locations in geographic information systems. Tominski and Schulz [47] propose the Great Wall of Space-Time, which extrudes a piece-wise linear spatial trajectory along the time dimension.

3 Method Overview

Our method starts with an ensemble $\{\mathbf{s}_1^{[1;m]},\dots,\mathbf{s}_n^{[1;m]}\}$ of n 2D time-dependent scalar forecast fields, where each forecast predicts a physical quantity such as geopotential height at consecutive time-step $t_i, i=1,\dots,m$, over a certain period. Subscripts and superscripts, respectively, denote the ensemble member and time-steps over which the dynamics are considered. For a prescribed forecast interval $[t_a;t_b]$, the sub-sequences $\mathbf{s}_1^{[a;b]},\dots,\mathbf{s}_n^{[a;b]}$ are treated as n single elements. For a given value ν, the iso-contours where the forecast fields of these sub-sequences take on ν are clustered into k clusters. k, a and b are given by the user. The iso-contours in the selected sub-sequences are clustered so that those having similar geometry *and* similar motion trajectories fall into the same groups.

Starting with the initial k clusters, our method proceeds backward in time and reconsiders this clustering at every time-step $t_i, i=(a-1),\dots,1$ with respect to the iso-contours of ensembles $\mathbf{s}_1^{[i,i]},\dots,\mathbf{s}_n^{[i,i]}$. If the dissimilarity of two clusters falls below a certain threshold at time-step i, the two clusters are merged and in the next iteration the reduced set of clusters is considered.

Even though many different clustering algorithms can be be used for creating the cluster hierarchy, we decided to use the method proposed by Ferstl et al. [11, 10]. It uses Agglomerative Hierarchical Clustering (AHC), in combination with a similarity metric for iso-contours that is based on a principal component transformation of signed distance fields in which these contours are the level-0 sets. The advantage of this method is twofold: Firstly, it comes with an abstract cluster representation showing the variability of contours per cluster. Secondly, it can be used efficiently to determine a median contour per cluster. Both issues are important in our approach for encoding the cluster hierarchy visually and, thus, to overcome the mentioned limitations of spaghetti plots. A detailed description of the cluster approach follows in the next section.

By using the information encoded in the hierarchical cluster tree, in combination with abstract visual cluster representations, we provide three different options for analyzing the contour data: a) The user can interactively animate through the time-steps and let the contours or clusters at every time-step being visualized. b) From the cluster medians a continuous space-time surface can be computed and visualized for each of the k initial clusters. c) A plot of stacked time-cuts showing the abstract cluster representation on a subset of time-steps can be selected. The work flow and visualization options provided by our approach are illustrated in Fig. 3.

4 Time-Hierarchical Clustering

We start with an ensemble of n scalar forecast fields evolving over a sequence of m time-steps. The general goal of our method is to compute a clustering of the iso-contours in these fields to a fixed iso-value ν. This clustering should reflect the diverging nature of the weather forecast ensemble.

In general it is not possible to find a single clustering that is valid at all time-steps. To account for this, we let the user select a time-window of interest comprising all time-steps in the interval $[t_a,t_b]$ ($t_a=t_b$ is allowed). We then perform an initial clustering of the iso-contours according to a similarity of the contours "averaged" over this time-window. Based on the initial clustering, our basic approach is to go through the sequence of time-steps in reversed order, starting at time t_j-1, and recursively merge pairs of clusters as soon as their similarity exceeds a certain threshold. Because we do not allow clusters to split or individual members to change their cluster, this yields a tree-like hierarchy of clusters, where each merge is associated to a specific time-step. This is illustrated in Fig. 4.

4.1 Initial Clustering and Contour Variability Plots

To analyze the spatial variability of a set of iso-contours, Ferstl et al. [11] represent the contours by signed distance functions (SDFs).

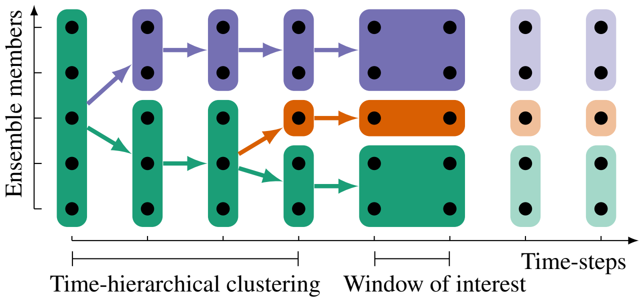

Fig. 4. Time-hierarchical clustering: Based on an initial clustering of the ensemble members in a user-selected window of interest, we build a hierarchy of clusters by merging them in time-reversed order. All later time-steps get assigned the clustering from the window of interest.

Let $\mathbf{s}_1,\ldots,\mathbf{s}_n \in \mathbb{R}^M$ be a set of scalar fields (defined on a grid with M vertices) in which the iso-contours of interest are given implicitly via the iso-value ν. For each $\mathbf{s}_i$, a signed distance transform according to the given iso-value is performed which yields corresponding SDFs $\mathbf{d}_i \in \mathbb{R}^M$. Each input contour i is now represented by a SDF $\mathbf{d}_i$, which can be interpreted as a high-dimensional point in $\mathbb{R}^M$, and, hence, the input set of iso-contours can be treated as an M-dimensional point cloud. In this representation, standard clustering methods can be applied to find clusters of iso-contours. For instance, AHC based on Euclidean distances and average linking, followed by an automated guess for an optimal number of clusters using the L-Method [41]. Furthermore, a principal component analysis of the point cloud can be performed first, to reduce the dimensionality of the space in which clustering is performed from M to $N << M$, yet in our application we omit this step because it does not improve performance.

Based on a given clustering of SDFs, a median contour can be computed very efficiently. Let $C \subseteq \{1,\ldots,n\}$ denote a cluster of SDFs, given as a point cloud in M-dimensional space. The so-called *geometric median* is then computed as the point which has the least sum of squared distances to all other points in the cluster (and generally does not coincide with an existing point). Given this median point, respectively the corresponding SDF, the *median contour* is implicitly given as its zero-contour.

Furthermore, as proposed by Ferstl et al., a so-called *contour variability plot* can be obtained by drawing, for each cluster of contours, a region, called *band*, which indicates the spatial standard deviation of the contours in the cluster. The band to a given number $\alpha > 0$ of standard deviations is obtained as the region enclosed between the zero-contours of the two artificial SDFs

$$\mathbf{d}_{1,2} := \underset{i\in C}{\operatorname{mean}}\{\mathbf{d}_i\} \pm \alpha \cdot \underset{i\in C}{\operatorname{std}}\{\mathbf{d}_i\}, \qquad (1)$$

where "mean" and "std" are operators which compute the component-wise mean values and standard deviations, respectively. Intuitively, since each component of a SDF $\mathbf{d}_i$ corresponds to a grid point, the band contains all grid points at which the value zero is within α standard deviations of all occurring values, i.e., grid points which are "likely" to be close to one of the given contours. In the current work, we make use of variability plots to generate an abstract cluster representation that can be used in our time-hierarchical visualization approach. We use $\alpha = 1$ in all of our examples.

4.2 Agglomerative Hierarchical Clustering

AHC is one of the most commonly used clustering methods (for a general overview of clustering techniques let us refer to the overview article by Jain [21]). Given n observations $\mathbf{x}_1,\ldots,\mathbf{x}_n$, AHC builds a hierarchy of clusters (an unbalanced binary tree) from which clusterings with different numbers of clusters can be derived. First, each observation is put into a separate cluster of cardinality one, and the hierarchy is then built by repeatedly merging pairs of similar clusters until all points are contained in a single cluster. To decide which clusters are merged in every iteration, AHC uses a distance metric and a so-called linkage criterion.

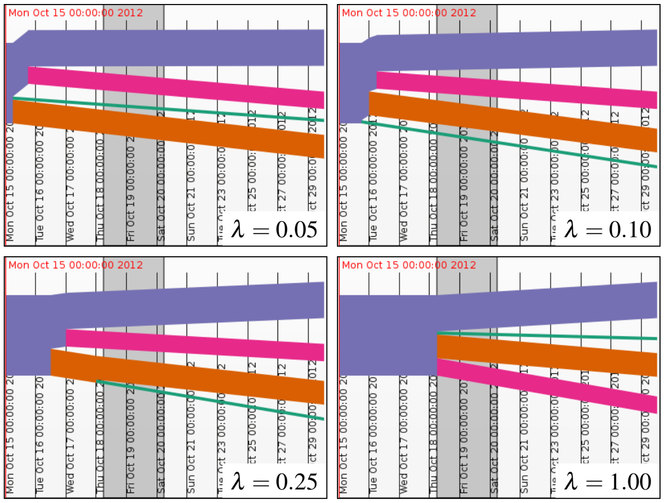

Fig. 5. Effect of different thresholds $\tau = \lambda \cdot \delta$ on the time-hierarchy of clusters for the scenario shown in Fig. 11. The horizontal axis corresponds to time, the gray region shows the window of interest ($= [t_a;t_b]$) over which the initial clustering is performed and the width of the colored bars corresponds to the cardinality of the clusters.

The distance metric determines the similarity between individual observations and can be generically specified through a distance matrix $D \in \mathbb{R}^{n\times n}$, where each entry $(D)_{ij}$ indicates the distance, or dissimilarity, between observations i and j. Throughout this work, we use the Euclidean distances $(D^t)_{ij} = \left|\left|\mathbf{s}_i^{[t;t]} - \mathbf{s}_j^{[t;t]}\right|\right|$ between the SDFs of corresponding iso-contours as distance metric. If an interval $[t_a;t_b]$ consisting of multiple time-steps is considered, we use the "mixed" Euclidean distances

$$\left(D^{[a;b]}\right)_{ij} = \sqrt{\sum_{t=a}^{b} \left|\left|\mathbf{s}_i^{[t;t]} - \mathbf{s}_j^{[t;t]}\right|\right|^2}.$$

The linkage criterion, on the other hand, specifies how distances between clusters are determined from the distances $(D)_{ij}$, such that in every iteration the pair of clusters with the smallest distance can be determined for merging. More precisely, given a clustering of $\{1,\ldots,n\}$ into k disjoint subsets, the linkage criterion defines inter-cluster distances as a linkage matrix $L \in \mathbb{R}^{k\times k}$ which is a function of the point-wise distances given in D. For practical uses, linkage criteria are typically expressed through a recursive update rule. Starting with $L := D$, this update rule specifies how the distances in L have to be adapted when two clusters are merged. When two disjoint clusters $I,J \subset \{1,\ldots,n\}$ are combined into a new cluster $I\cup J$, the corresponding two rows and two columns in L are collapsed and the new entries—the distances of the new cluster to other clusters $K \subset \{1,\ldots,n\}$— are determined from the distances between I, J and K:

$$(L)_{I\cup J,K} = f\Big((L)_{I,K},(L)_{J,K},(L)_{I,J}\Big).$$

In this work we use linking according to Ward's method [20] because it produces more equally sized clusters than average linking as used by Ferstl et al. [11]. This linkage criterion seeks to minimize the sum of within-cluster variances, and the corresponding update rule is:

$$(L)_{I\cup J,K} = \frac{(|I|+|K|)\cdot(L)_{I,K} + (|J|+|K|)\cdot(L)_{J,K} - |K|\cdot(L)_{I,J}}{|I|+|J|+|K|}.$$

Note that the choice of linkage criterion is application dependent and has a strong influence on the resulting clusterings. Hence, different criteria might be preferable in other cases (e.g., average, single or even complete linking).

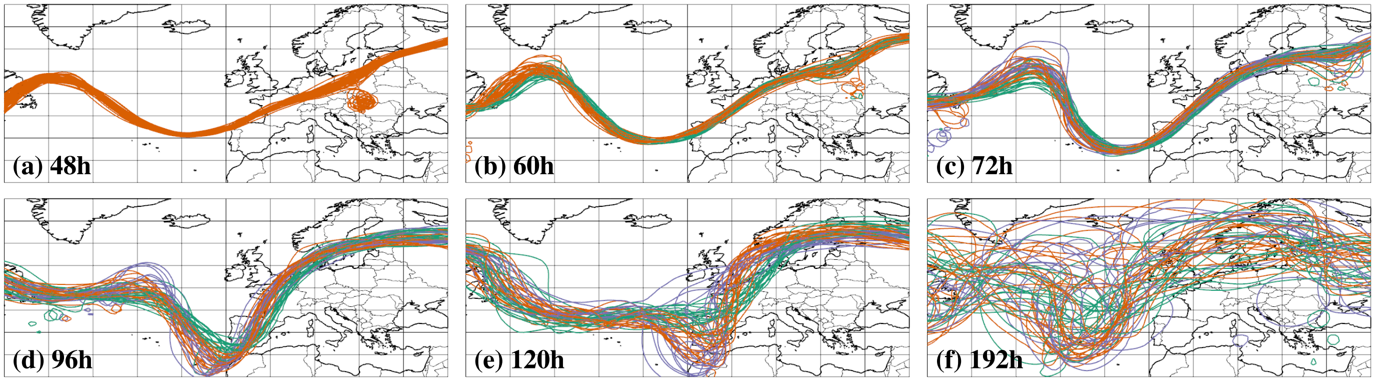

Fig. 6. Spaghetti plots of an ensemble of 5600 m geopotential height contour lines at 500 hPa. Each plot shows the lines of all 51 members of the ECMWF ENS forecast, colored by cluster membership. The forecast from 00:00 UTC 15 October 2012, valid at the indicated lead times, is shown.

4.3 Generation of the time-hierarchy

To generate the required time-hierarchical clustering, we follow the idea of AHC and use a linkage criterion to determine when clusters should be merged. The major difference is that now we are considering multiple time-steps and that the distances between the observations are time-dependent, i.e., there is a different distance matrix $D^t \in \mathbb{R}^{n \times n}$ in every time-step.

We start by defining an inter-cluster distance threshold τ, which is used to decide when clusters will be merged. Since different distance metrics and inputs can result in distance values at vastly different scales, we make this threshold depend on the initial clustering in our time-window of interest. Given the corresponding distance matrix $D^{[a;b]}$ that was used to generate the initial clustering with k clusters, we compute the corresponding linkage matrix $L^{[a;b]} \in \mathbb{R}^{k \times k}$. Let δ be the smallest non-zero entry of $L^{[a;b]}$. Then, in the sense of AHC, δ is the minimum distance between the clusters in the initial clustering, and we choose the inter-cluster distance threshold relative to δ as $\tau := \lambda \cdot \delta$ with $\lambda \in]0;1]$. The factor λ can be used to control how fast, or aggressive, our time-hierarchical clustering merges clusters. Larger λ will cause clusters to be merged more aggressively, while smaller λ will defer the merging of clusters to earlier time-steps.

Once τ has been selected, we iterate through the sequence of time-steps in reversed order, starting at time-step $a-1$, and try to merge clusters as soon as their distance according to the linkage criterion becomes smaller than τ. In every time-step t, similar to before, we use the current clustering with k clusters and the current distance matrix D^t to compute the corresponding linkage matrix $L^t \in \mathbb{R}^{k \times k}$. If any non-zero entry in L^t is smaller than τ, we merge the two corresponding clusters and update L^t (analogously to regular AHC). We repeat this procedure in time-step t until no more merges can be performed with a distance smaller than τ, and then proceed to time-step $t-1$. After processing the first time-step, our time-hierarchical clustering is finished. Note that, in general, the resulting hierarchy is not a binary tree because more than one merge can occur in one time-step, and that there is no guarantee that all clusters are merged into a single cluster after processing the first time-step. The number of clusters in the first time-step, however, can be influenced by changing λ. The effect of different choices for λ on the time-hierarchy of clusters is illustrated in Fig. 5.

5 Space-time Visualization

Given the time-hierarchical clustering of iso-contours, the corresponding cluster tree has to be visualized so that both aspects, the time evolution of single clusters and the splitting of clusters over time, can be conveyed effectively. This is difficult to achieve, since it requires to show when and where variations occur that cause clusters to change. In principle, common techniques including juxtaposition and animation of spaghetti plots of the contours in each time-step can be applied, further enhanced by coloring the contours according to their cluster membership.

Fig. 6 shows juxtaposes, i.e., side-by-side placement in one view, of spaghetti plots of the 51 iso-contours in a weather forecasts at 9 consecutive time steps, colored according to the proposed time-hierarchical clustering. One can see immediately that from this kind of visualization it is difficult to grasp the major spatial shape and variation of the clusters, their spatial changes over time, and the temporal split events. Animation, on the other hand, besides its known limitations, fails especially when a sequence of static spaghetti plots is displayed. This is due to the large number of contour lines that need to be memorized from image to image and the often rather chaotic appearance of the plots.

In a first attempt to improve the visualization, one can replace the spaghetti plots in each time-step by an abstract spatial cluster representation, for instance, via contour boxplots [48], variability plots [11], or any other suitable abstraction. As seen in Fig. 7, this helps to better convey the clusters' shapes, variations, and splits. Yet it is still difficult to perceive the relationships between the structures in time-steps which are not placed close to each other, a common problem of juxtaposition as pointed out by Gleicher et al. [14].

5.1 Space-time cubes

To overcome the limitations of juxtaposition and animation, we propose to visualize the time evolution of clusters via new techniques that built upon the concept of *space-time cubes* [16, 24]. Space-time cubes define a 3D orthogonal coordinate system, where two dimensions refer to space and the third dimension refers to time. Spatiotemporal data points are displayed in this coordinate system to reveal spatiotemporal patterns in the data. In an between-subjects experiment [25] it was demonstrated that users, when using space-time cubes, needed only half of the time that was required when using alternative techniques, to answer complex questions requiring an overall understanding of the spatiotemporal structure of the data.

5.1.1 Space-time cluster surfaces

Firstly, we propose the visualization of the temporal cluster evolution by connecting iso-contours across time to form a *space-time cluster surface*. The approach has similarities to the Great Wall of Space-Time by Tominski and Schulz [47], yet it does not extrude a piecewise linear base contour along constant time trajectories, but forms a smooth surface from arbitrarily given iso-contours at every time-step. To build the surface, we first compute in every time-step and for every cluster a representative median contour (see Section 10). Other representatives, like the mean contour, are possible, yet since a Gaussian distribution assumption is not always justified for the spread of contours belonging to one cluster, we decided to use solely median contours in this work.

Instead of first extracting polygonal representations of the median contours and constructing the cluster surfaces from them, we propose a different approach: For every cluster, we take the signed distance fields of the median contours and stack them together consecutively

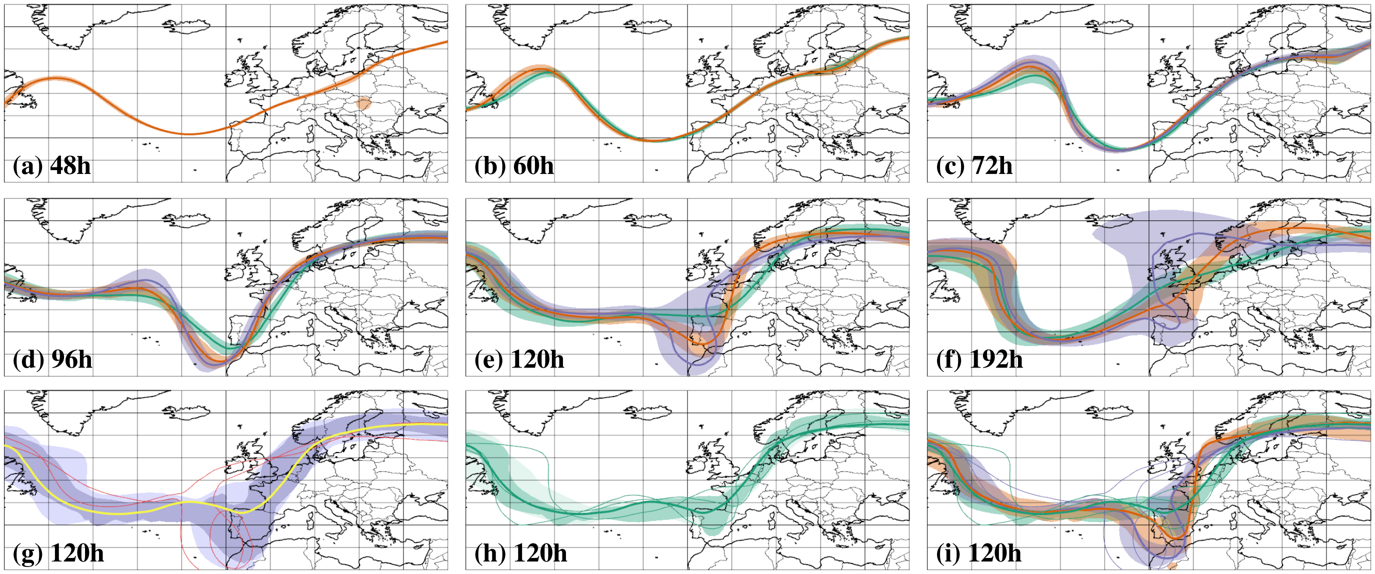

Fig. 7. Analoguous to Fig. 6. **(a-f)** Contour variability plots proposed by Ferstl et al. [11]. **(g-i)** Contour boxplots as proposed by Whitaker et al. [48]: **(g)** Contour boxplot computed from all 51 ensemble members, **(h)** from a single cluster, and **(i)** one contour boxplot computed from each cluster (100% data depth bands omitted for simplicity).

with respect to time to form a space-time distance volume. By drawing the level-0 sets in these volumes using volumetric ray-casting and tri-linear interpolation, temporal interpolation between contours is performed and a smoothly varying cluster surface is rendered. A drawback of this approach is that it can result in an inconsistent surface, for instance, when a contour splits into two disconnected contours from one time-step to the next. However, if the time step of the numerical weather prediction output is small enough for the scale of the considered atmospheric phenomenon, such inconsistencies are rare. As an example, Fig. 8(a) shows a space-time cluster surface for a single cluster evolving over time.

To also reveal the spread of the contours belonging to a cluster, the points of the space-time cluster surfaces are colored in the following way: At every surface point, we compute the standard deviation of the signed distance values of all contours, i.e., their distance fields, belonging to the cluster. The standard deviation is then mapped to color, as shown in Fig. 8(b). In the shown example, the coloring shows clearly the increasing spatial spread of the cluster members with advancing time. Also, the plot shows increased spread along the trough which is moving eastwards.

An apparent limitation of cluster surfaces are occlusions which are introduced especially when a highly curved cluster surface consisting of multiple layers is visualized. It is clear that in such a case it is not always possible to perceive from the surface the overall shape of the cluster. On the other hand, this problem can at least partly be alleviated by rendering the surface on the GPU. Even for high resolution fields, GPU volume ray-casting can achieve very high frame rates, enabling an interactive navigation and a detailed exploration of the cluster surface from different perspectives. In combination with clipping planes and by using transparency and shading effect to emphasize specific features, a powerful visual encoding of a time-varying cluster of isocontours is given.

In principle, the entire cluster tree can be rendered in one single image containing all individual cluster surfaces, via multi-parameter volume rendering, i.e., by testing at every sample point along the rays against the stacked signed distance volumes of all clusters, and rendering simultaneously the implicitly given cluster surface in each volume. The resulting visualization is shown in Fig. 8(c), and it immediately shows the expected problem of this kind of visualization: Since cluster surfaces overlap and occlude each other, it is hardly possible to grasp the relevant spatial and temporal changes in one single view. It can nevertheless be perceived clearly, how the clusters diverge from each other over time. This is indicated by the fan-like appearance of the surfaces tree, indicating the fanning out of the cluster medians with increasing time.

5.1.2 Stacked time-cuts

To overcome the restriction of space-time cluster surfaces to only one single cluster, we propose an alternative visualization which also builds upon space-time cubes. The major problem of cluster surfaces as proposed is that relevant geometric features can be occluded by other surface parts and spatial reference is increasingly lost with advancing time, i.e., increasing distance of geometric features to the base plane. To alleviate these problems, we first restrict the visualization of the cluster information to a discrete set of time-steps, so-called time-cuts, by rendering these cuts as a set of slices stacked on top of each other (see Figs. 10+11). The advantage of such a slice-based visualization is that it enables an almost un-occluded view on the clusters in each slice, at the same time exploiting the humans's perceptional capabilities to form a mental image of the information in-between. Similar to space-time cluster surfaces, however, care needs to be taken to choose the time step between two slices small enough for the scale of the considered phenomenon.

With space-time slices, however, it is still difficult to identify the spatial relationships between features at different times, because no common points of orientation are given in the slices. Therefore, we slightly modify and enhance the plot by a) visualizing in each slice a map showing the ground over which the space-time cube is built, and b) letting the user place and move interactively so called vertical drop lines at reference locations. In combination with rendering the cube's edges, drawing silhouettes around the slices, and color coding cluster and ground information differently, a significantly enhanced localization of features and their spatial relationships is obtained.

6 Results

To demonstrate the practical application of the proposed methods, we discuss two real-world weather forecasting cases that occurred during an atmospheric research campaign ("T-NAWDEX-Falcon", cf. [43]) in 2012 and analyze the corresponding time-hierarchical clusterings. The cases were also previously used by Ferstl et al. [11] and Rautenhaus et al. [40]. Forecast data are obtained from the ECMWF Ensemble Prediction System (ENS). The ensemble comprises an unperturbed control run (i.e., started from the "best" initial conditions) and 50 perturbed members (e.g., [26]). Our example region covers the North At-

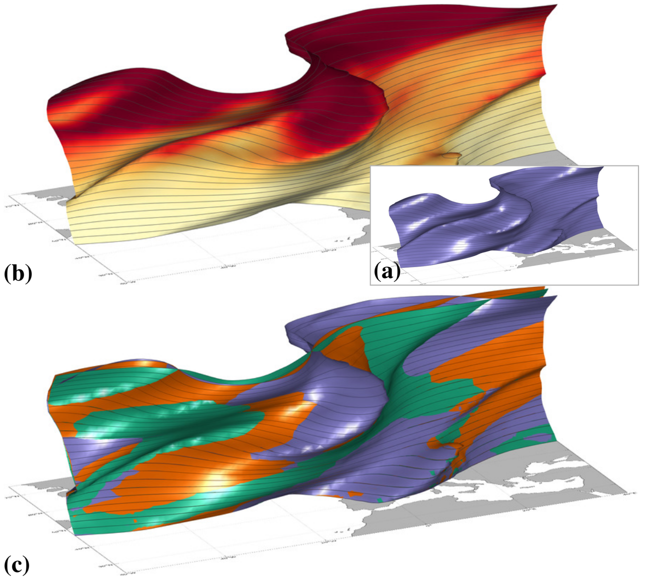

Fig. 8. Space-time cluster surfaces: a) Single cluster median. b) Same as a) but with color coded standard deviation. c) Simultaneous visualization of all time-hierarchical clusters.

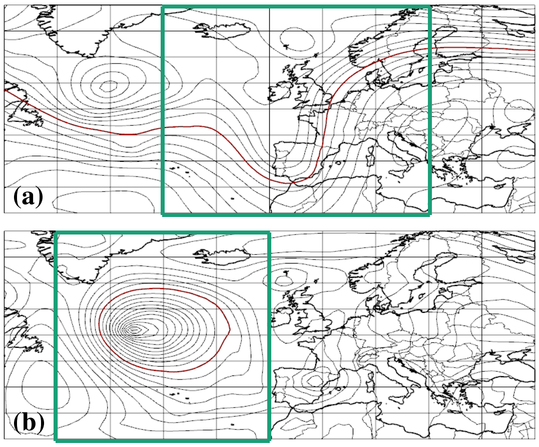

Fig. 9. Context for the two real-world examples. **(a)** Geopotential height contour lines of the ECMWF ensemble control forecast at 500 hPa. The forecast from 00:00 UTC 15 October 2012, valid at 12:00 UTC 19 October 2012, is shown. The 5600 m contour line is highlighted in red. The green box shows the selected region of interest. **(a)** The same as (a) but at 925 hPa, showing the forecast from 00:00 UTC 17 October 2012, valid at 00:00 UTC 21 October 2012. The 680 m contour line is highlighted in red.

lantic and Europe and encompasses 101 × 41 × 62 grid points. Data are available on vertical levels of constant pressure from the TIGGE archive [45]. For the present study, we use forecasts from 00:00 UTC 15 October 2012 and from 00:00 UTC 17 October 2012.

Visualization methods including contour variability plots, time-hierarchical cluster tree view and stacked time-cuts have been implemented in the open-source ensemble visualization tool Met.3D [40]. Figure 1 shows a screenshot of the system in use. The user is able to interactively navigate through time and within the displayed scenes in real-time. Cluster correspondence of the individual ensemble members is pre-computed. For the computation of the initial clustering, our implementation uses the MATLAB implementation of AHC. The computation of SDFs via fast marching [44] and the time-hierarchical clustering use a custom implementation. All presented results were generated on a standard desktop PC (Intel Xeon X5675 processor with 6 × 3.0 GHz, 8 GB RAM and an NVIDIA Geforce GTX 680). The time required to compute a time-hierarchical clustering is dictated by the computation of SDFs and takes about 1.5 s for both shown cases (the generation of the initial clustering and the time-hierarchy only take about 30ms). Rendering times for the horizontal sections in Met.3D are on the order of a few ms (cf. Table 2 in [40]).

6.1 Example 1: Trough

Our first example examines the temporal development of an upper-level low-pressure trough over the North Atlantic and Europe. We assume the following scenario: On Monday, 15 October 2012, the user (forecaster) is interested in the large-scale atmospheric development towards the end of the week (cf. the scenario described in [43]). The geopotential height field at 500 hPa is a common product to judge such developments (e.g., [1]). Figure 9a shows the forecast geopotential height field of the control forecast, valid at 12:00 UTC 19 October 2012. A distinct trough approaches Europe, extending from Ireland over Spain to Morocco (cf. Fig. 2 in [43]). We select the 5600 m contour as a representative line for the situation to judge the uncertainty in the ensemble (red line in Fig. 9a). To apply the time-hierarchical clustering algorithm, we define a region covering 30° W to 20° E and 30° N to 70° N (green box in Fig. 9a) and a 24-hour time window starting at 12:00 UTC 19 October 2012 as the region of interest.

Fig. 6 shows spaghetti plots of a number of time steps of the forecast, colored by cluster membership. The plots up to a forecast lead time of 120 h clearly show the approximate development of the 5600 m contour, at 120 h an increased uncertainty in the region of interest can be observed. At longer lead times, the plot quickly becomes illegible. Also, as noted in Sec. 5, the spatial shape of the clusters, their temporal development, and, in particular, the split events are hard to discern.

Our proposed alternative visualizations, the space-time cluster surfaces and the stacked time-cuts with time-hierarchical clustering, are shown in Figs. 8 and 10. If the user is interested in a particular cluster, Fig. 8a shows the space-time cluster surface. The depiction is similar to a 2D Hovmöller diagram [13] and at a glance provides a qualitative depiction of the trough's eastward movement with time. However, in contrast to a 2D Hovmöller diagram, the 3D display frees the color channel to display the corresponding cluster's standard deviation (Fig. 8b). As discussed for the spaghetti plots, this depiction as well highlights that uncertainty first increases around the trough.

The cluster tree (Fig. 10) provides a compact summary of the clustering computed from the selected region of interest (highlighted in gray). In the selected time window, the algorithm distinguishes three clusters of roughly similar size (with the purples cluster slightly smaller). In the first 60 h of the forecast, the ensemble members are similar to each other (all in the orange cluster), at 60 h and 66 h lead time, the members split into the final clusters. In the stacked time-cuts, the temporal development of the clusters is immediately visible. A time step of 12 h has been chosen between consecutive slices, small enough to capture the development of the considered large-scale phenomenon. As a spatial reference, we have placed a vertical pole west of Portugal (the pole can be interactively moved by the user). The visualization clearly shows how and when the forecast splits into three scenarios that predict troughs of different strength in the region of interest (purple cluster – strong trough; green cluster – weak trough; orange cluster – in between). Hence, the information of interest (the possible scenarios for the temporal development of the trough) can very quickly be obtained from a single image.

6.2 Example 2: Low pressure system contours

Our second example focuses on the strong low-pressure system which is visible south of Greenland in Fig. 9a (former Hurricane Raphael; cf. [43] and Fig. 10b in [11]). Closer to the surface at 925 hPa and in the forecast initialized 48 h later (00:00 UTC 17 October 2012), the system can be represented by the 680 m contour line. Figure 9b shows the control forecast valid at 00:00 UTC 20 October 2012, along with a spatial region of interest centered around the system (50° W to 10° W and 30° N to 70° N). As the time window of interest, we define a 48-hour interval starting at 06:00 UTC 20 October 2012.

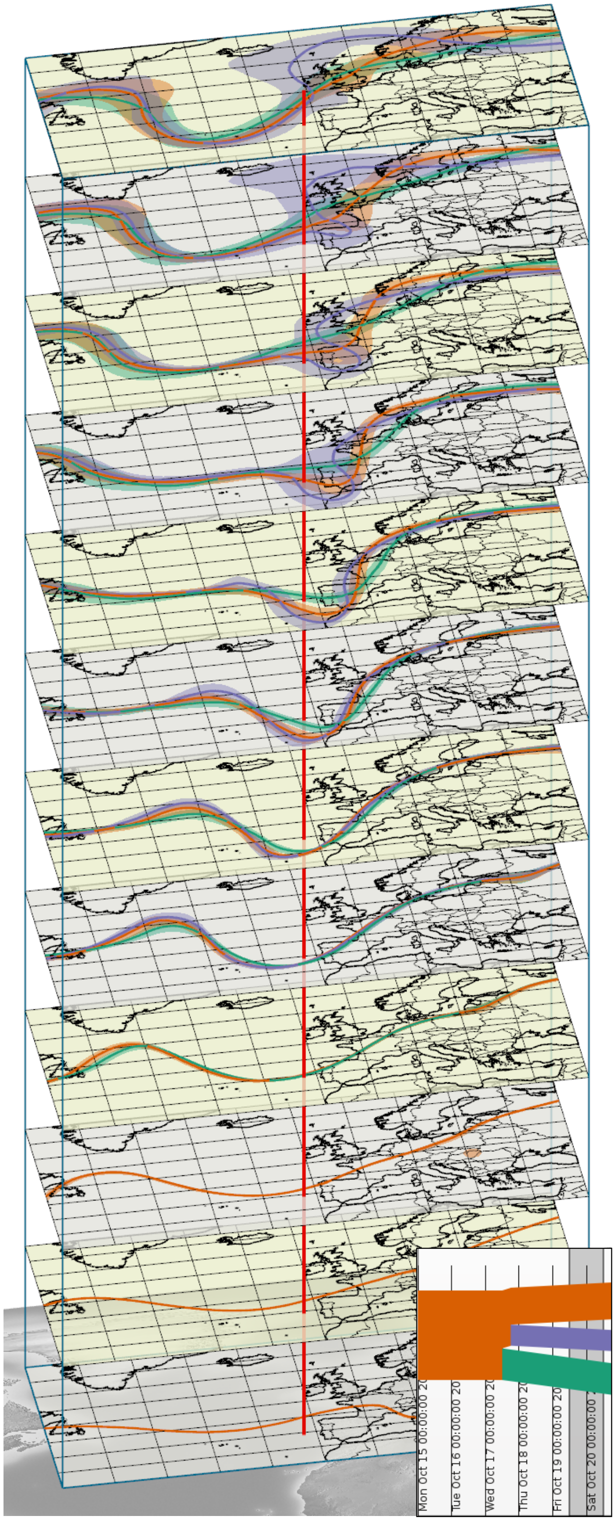

Fig. 10. Stacked time-cuts with variability plots of an ensemble of 5600 m geopotential height contour lines at an elevation of 500 hPa. The ECMWF ENS forecast from 00:00 UTC 15 October 2012, valid at 00:00 UTC 16 October 2012 (lowest slice, 24 h forecast lead time) to 12:00 UTC 21 October 2012 (uppermost slice, 168 h forecast lead time), is shown. Slices are plotted at 12 h intervals. Note how the ensemble members cluster into forecasts that predict a trough of varying strength. A vertical pole (red) provides a spatial reference for the temporal development of the trough features. The inset shows the corresponding time-hierarchical cluster tree (the temporal region of interest is highlighted in gray). Compare to Fig. 8 in [11].

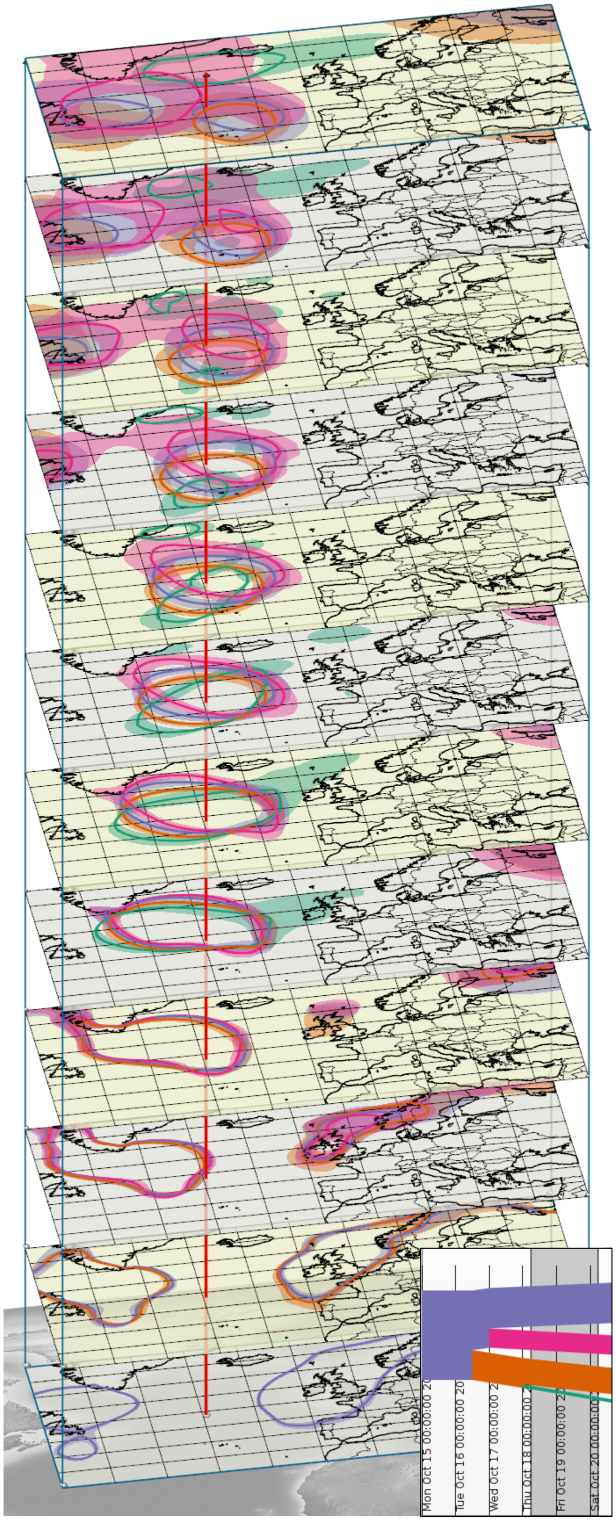

Fig. 11. Stacked time-cuts with variability plots and time-hierarchical cluster tree as in Fig. 10 but showing an ensemble of 680 m geopotential height contour lines at an elevation of 925 hPa. The ECMWF ENS forecast from 00:00 UTC 17 October 2012, valid at 00:00 UTC 18 October 2012 (lowest slice, 24 h forecast lead time) to 12:00 UTC 23 October 2012 (uppermost slice, 168 h forecast lead time), is shown. Slices are plotted at 12 h intervals. Compare to Fig. 10b in [11].

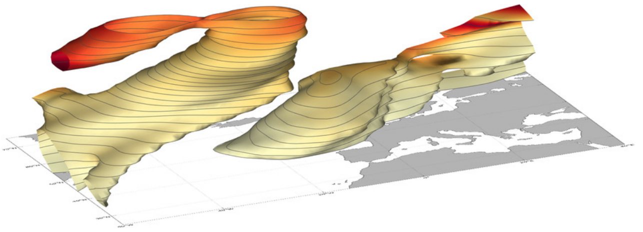

Fig. 12. Space-time cluster surface of the purple cluster in Fig. 11, colored by standard deviation of signed distance values.

Figure 1 shows a screenshot of our system during the analysis of the forecast case, Fig. 11 shows stacked time-cuts as in Fig. 10. In this example, the clustering algorithm distinguishes four forecast scenarios for the region of interest, this time with differing cluster sizes. In particular, note that the green cluster contains only very few members, while the majority of members (i.e., the most likely scenario) is represented by the purple cluster. Again, the stacked time-cuts provide a compact visual summary of the temporal development of the forecasts in a single image. We see that the four clusters mainly differ in shape and orientation in the temporal region of interest, at later time steps they are also shifted in space and differences become larger. The space-time cluster surface of the purple cluster is shown in Fig. 12. In this view, the continuous time evolution of the cluster median it is easier to see. In particular, the approach of a second depression towards the end of the forecast period is highlighted. The red color, however, indicates the high associated uncertainty.

6.3 Characteristics of time-hierarchical clustering

To analyze characteristics of the proposed time-hierarchical clustering technique, we computed average inter-cluster and intra-cluster distances for each time-step t of both example cases. The average inter-cluster distance is computed as the average of all entries in D^t which correspond to a distance between members of two different clusters. The average intra-cluster distance for each cluster is computed as the average of all non-zero entries in D^t which correspond to a distance between two members of the respective cluster.

Figure 13 shows the temporal evolution of both measures. As expected from the assumption of increasing ensemble spread with time, the average inter-cluster distance increases as well. The only exception can be observed after 54 h lead time in example 2, where the clusters slightly approach each other before starting to diverge again.

The intra-cluster distances also increase with time over most parts of the forecasts, however, distances slightly decreasing with time can be observed as well. If both splits and merges were allowed in our algorithm, decreasing distances (with time) could cause clusters to merge again. This could happen, e.g., in Fig. 13a, where a "temporary" split might occur starting from 36 to 42 h lead time and ending from 48 h to 54 h lead time. We note that since our algorithm moves backward in time, the identified splits (in forward time direction) are always the last possible splits when compared to the case that re-merging clusters are allowed. The algorithm could also be modified to always find the first split with increasing time.

For our examples, we observe in all timesteps that the average inter-cluster distance is greater or equal than (most of) the average intra-cluster distances. However, the strength of this characteristic decreases with increasing time difference to the time window of interest.

7 Limitations and Future Work

Underlying our approach is the assumption that there is a monotonic increase in the number of clusters over time and that merge events do not occur. Thus, given the initial clustering regarding the selected time window, the clusters which are derived via our hierarchical merging backwards in time cannot always represent the contour distribution at earlier time steps equally well. The time-hierarchy rather shows how the initial clusters develop prior to the selected time window. Because of this, misleading results can be generated for ensembles not showing the specific diverging nature which we assume. In the future we plan to extend our approach with respect to this limitation, by considering split and merge events like the ones discussed in the previous section, and by providing options to analyze the stability of the cluster evolution over time. Secondly, to generate the initial clustering, we will analyze the effect of using alternative clustering approaches that work on scalar fields instead of on iso-contours, such as the clustering used at ECMWF [9]. Finally, we will investigate how to embed space-time cluster surfaces into stacked representations to enable a better estimation of the cluster evolutions in-between the visualized time-cuts.

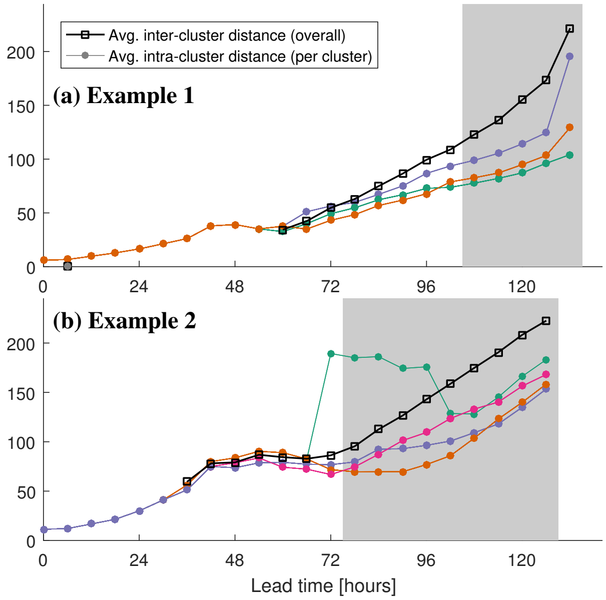

Fig. 13. Average inter- and intra-cluster distances for the examples shown in **(a)** Fig. 10 and **(b)** Fig. 11. Average inter-cluster distances can only be computed in timesteps with more than one cluster.

8 Conclusion

In this work we have presented a new approach for visualizing the spatial and temporal evolution of iso-contours in ensembles of 2D time-varying scalar fields. We have achieved this via a specific time-hierarchical clustering of the initial set of contours, and the conversion of the contours per cluster into an abstract representation showing the major trend and spread of the cluster. We have introduced novel visualization approaches for analyzing the evolution of single clusters over time, via space-time cluster surfaces, and the simultaneous depiction of all clusters in a spatio-temporal context, via stacked time-cuts. In combination with additional geo-references, an intuitive understanding of the evolution of the clusters over time is enabled.

Acknowledgements

Access to ECMWF prediction data has been kindly provided in the context of the ECMWF special project "Support Tool for HALO Missions". This work was supported by the European Union under the ERC Advanced Grant 291372—SaferVis: Uncertainty Visualization for Reliable Data Discovery.

References

[1] C. D. Ahrens. *Meteorology Today*. Brooks Cole, 9 edition, July 2008. 7

[2] N. Andrienko, G. Andrienko, and P. Gatalsky. Visual data exploration using space-time cube. In *Proceedings of the 21st International Cartographic Conference*, pages 1981–1983, 2003. 3

[3] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A review of temporal data visualizations based on space-time cube operations. In *Eurographics Conference on Visualization*, 2014. 3

[4] P. Bauer, A. Thorpe, and G. Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, Sept. 2015. 1

[5] G.-P. Bonneau, H.-C. Hege, C. Johnson, M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. Overview and state-of-the-art of uncertainty visualization. In *Scientific Visualization*, pages 3–27. Springer, 2014. 2

[6] S. Bruckner and T. Möller. Isosurface similarity maps. *Computer Graphics Forum*, 29(3):773–782, 2010. 2

[7] H. Carr, D. Brian, and D. Brian. On histograms and isosurface statistics. *IEEE Trans. Visual. and Comp. Graphics*, 12(5):1259–1266, 2006. 2

[8] A. Denton. Clustering of time series data. *Encyclopedia of Data Warehousing and Mining*, pages 258–263, 2005. 3

[9] L. Ferranti and S. Corti. New clustering products. *ECMWF Newsletter*, 127:6–11, 2011. 1, 2, 9

[10] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, 2016. 3

[11] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Visual analysis of spatial variability and global correlations in ensembles of isocontours. In *Computer Graphics Forum (Proc. EuroVis)*, 2016. to appear. 1, 2, 3, 4, 5, 6, 7, 8

[12] A. Fofonov, V. Molchanov, and L. Linsen. Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views. *IEEE Trans. Visual. and Comp. Graphics,*, to appear, 2016. 2

[13] I. Glatt, A. Dörnbrack, S. Jones, J. Keller, O. Martius, A. Müller, D. H. W. Peters, and V. Wirth. Utility of hovmöller diagrams to diagnose rossby wave trains. *Tellus A*, 63(5):991–1006, Oct. 2011. 7

[14] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. Hansen, and J. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011. 1, 5

[15] C. Goutte, P. Toft, E. Rostrup, F. Å. Nielsen, and L. K. Hansen. On clustering fMRI time series. *NeuroImage*, 9(3):298 – 310, 1999. 3

[16] T. Hägerstraand. What about people in regional science? *Papers in Regional Science*, 24(1):7–24, 1970. 3, 5

[17] J. Heinrich and D. Weiskopf. State of the Art of Parallel Coordinates. In *Eurographics 2013 - State of the Art Reports*, 2013. 2

[18] M. Hlawatsch, P. Leube, W. Nowak, and D. Weiskopf. Flow radar glyphs—static visualization of unsteady flow with uncertainty. *IEEE Trans. on Visualization and Comp. Graph.*, 17(12):1949–1958, 2011. 2

[19] M. Hummel, H. Obermaier, C. Garth, and K. Joy. Comparative visual analysis of Lagrangian transport in CFD ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2743–2752, 2013. 2

[20] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Assoc.*, 58(301):236–244, 1963. 4

[21] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. 2, 4

[22] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, 2013. 2

[23] W. Kendall, J. Huang, and T. Peterka. Geometric quantification of features in large flow fields. *IEEE Computer Graphics and Applications*, 32(4):46–54, 2012. 2

[24] M. Kraak. The space-time cube revisited from a geovisualization perspective. In *Proceedings of the 21st International Cartographic Conference*, pages 1988–1996, 2003. 3, 5

[25] P. O. 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 Trans. on Visualization and Computer Graphics*, 15(4):696–702, 2009. 5

[26] M. Leutbecher and T. Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008. 1, 2, 6

[27] T. W. Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857 – 1874, 2005. 3

[28] A. L. Love, A. Pang, and D. L. Kao. Visualizing spatial multivalue data. *IEEE Computer Graphics and Applications*, 25(3):69–79, 2005. 2

[29] `www.spc.noaa.gov/exper/sref/`. 1

[30] T. Nocke, M. Flechsig, and U. Bohm. Visual exploration and evaluation of climate-related simulation data. In *Winter Simulation Conference 2007*, pages 703–711, 2007. 2

[31] S. Oeltze, D. J. Lehmann, A. Kuhn, G. Janiga, H. Theisel, and B. Preim. Blood flow clustering and applications in virtual stenting of intracranial aneurysms. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):686–701, 2014. 2

[32] M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2D vector field topology. *Computer Graphics Forum*, 29(2):347–356, 2010. 2

[33] T. Pfaffelmoser and R. Westermann. Visualizing contour distributions in 2D ensemble data. In *Short Paper Proc. EuroVis*, pages 55–59, 2013. 2

[34] H. Piringer, S. Pajer, W. Berger, and H. Teichmann. Comparative visual analysis of 2D function ensembles. *Computer Graphics Forum*, 31(3pt3):1195–1204, 2012. 2

[35] S. Policker and A. B. Geva. Nonstationary time series analysis by temporal clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 30(2):339–343, 2000. 3

[36] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum (Proc. of EuroVis)*, 29(3):823–831, 2010. 2

[37] K. Potter, P. Rosen, and C. R. Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. In *Uncertainty Quantification in Scientific Computing*, pages 226–249. 2012. 2

[38] P. S. Quinan and M. Meyer. Visually comparing weather features in forecasts. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):389–398, 2016. 3

[39] Y. Rathi, S. Dambreville, and A. Tannenbaum. Statistical shape analysis using kernel PCA. *Proc. SPIE*, 6064:60641B, 2006. 2

[40] M. Rautenhaus, M. Kern, A. Schäfler, and R. Westermann. Three-dimensional visualization of ensemble weather forecasts – Part 1: The visualization tool Met.3D (version 1.0). *Geoscientific Model Development*, 8(7):2329–2353, 2015. 6, 7

[41] S. Salvador and P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 576–584, 2004. 4

[42] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. J. Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16:1421–1430, 2010. 1, 3

[43] A. Schäfler, M. Boettcher, C. M. Grams, M. Rautenhaus, H. Sodemann, and H. Wernli. Planning aircraft measurements within a warm conveyor belt. *Weather*, 69(6):161–166, June 2014. 6, 7

[44] J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge Univ. Press, 1999. 7

[45] R. Swinbank, M. Kyouda, P. Buchanan, L. Froude, T. M. Hamill, T. D. Hewson, J. H. Keller, M. Matsueda, J. Methven, F. Pappenberger, M. Scheuerer, H. A. Titley, L. Wilson, and M. Yamaguchi. The TIGGE project and its achievements. *Bulletin of the American Meteorology Society*, 97(1):49–67, 2015. 7

[46] D. Thomas and V. Natarajan. Multiscale symmetry detection in scalar fields by clustering contours. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2427–2436, 2014. 2

[47] C. Tominski and H.-J. Schulz. The great wall of space-time. In *Vision, Modeling and Visualization*, 2012. 3, 5

[48] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2713–2722, 2013. 1, 3, 5, 6

[49] J. J. V. Wijk and E. R. V. Selow. Cluster and calendar based visualization of time series data. In *Proc. of the IEEE Symposium on Information Visualization*, pages 4–9, 140, 1999. 3

[50] D. S. Wilks. *Statistical Methods in the Atmospheric Sciences*. Academic Press, 3rd edition, June 2011. 1, 3

[51] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, 1996. 2

[52] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc. 18th Int. Conf. on Pattern Recognition*, pages 1135–1138, 2006. 2