

Visual Analysis of Multi-Parameter Distributions Across Ensembles of 3D Fields

Alexander Kumpf^{ID}, Josef Stumpfegger^{ID}, Patrick Fabian Härtl, and Rüdiger Westermann^{ID}

Abstract—For an ensemble of 3D multi-parameter fields, we present a visual analytics workflow to analyse whether and which parts of a selected multi-parameter distribution is present in all ensemble members. Supported by a parallel coordinate plot, a multi-parameter brush is applied to all ensemble members to select data points with similar multi-parameter distribution. By a combination of spatial sub-division and a covariance analysis of partitioned sub-sets of data points, a tight partition in multi-parameter space with reduced number of selected data points is obtained. To assess the representativeness of the selected multi-parameter distribution across the ensemble, we propose a novel extension of violin plots that can show multiple parameter distributions simultaneously. We investigate the visual design that effectively conveys (dis-)similarities in multi-parameter distributions, and demonstrate that users can quickly comprehend parameter-specific differences regarding distribution shape and representativeness from a side-by-side view of these plots. In a 3D spatial view, users can analyse and compare the spatial distribution of selected data points in different ensemble members via interval-based isosurface raycasting. In two real-world application cases we show how our approach is used to analyse the multi-parameter distributions across an ensemble of 3D fields.

Index Terms—Ensemble visualization, multi-parameter visualization, 3D rendering, distribution comparison, parallel coordinates

1 INTRODUCTION

IN many scientific fields like meteorology and computational fluid dynamics, numerical ensemble simulations are carried out with varying magnitudes of initial condition uncertainty, and by introducing uncertainty in the representation of certain physical processes. In such an ensemble, each simulation predicts possible states of physical quantities, aiming at a representative sampling of the physical phenomena that can occur. One of the major goals in ensemble visualization is to visually convey commonalities and differences between the ensemble members and, thus, to reveal the major trends and outliers in the simulation results.

While there is a considerable body of work related to the visual analysis of single-parameter ensembles [64], research dedicated to the analysis of multi-parameter ensembles, to the best of our knowledge, is rare. A multi-parameter ensemble is an ensemble of fields where at each domain point a set of physical variables, the so-called output parameters, is given. For instance, one of our use cases (Fig. 1) is an ensemble of 3D cloud simulations comprising 96 members, each consisting of 250 K data points with twelve different physical output parameters. The ensemble is generated by varying the input parameters of the simulation. These parameters are not considered in this work.

Previous works in multi-parameter data visualization, also termed multi-faceted, multi-field or multi-dimensional data visualization [38], [43], have focused on visualizing the relationships between the parameters in a single data set, for instance, via parallel coordinates [33], dimensionality reduction [6], [37], or pair-wise scatterplot matrices [8]. In principle, such techniques can also be used to analyse the parameter variations across an ensemble, by either juxtaposition of single member visualizations or combined visualization of multiple members. In our use case, however, even for the visualization of a single member these techniques alone are not feasible due to visual clutter and over-plotting, and a combined visualization that can convey relationships between members becomes increasingly challenging.

The number of entities that need to be considered when analyzing a multi-parameter data set can be reduced effectively via clustering in parameter space. By finding groups of data points with similar multi-parameter values, cluster-based visual analysis using graphical abstractions for sets of data points can be utilized. However, when clustering is used with an ensemble—by clustering each ensemble member individually and comparing the results [39]—differences in the number of clusters and their composition further complicate the analysis. Especially the use of approximate cluster matchings to compare different clusterings can result in misleading decisions.

To address these shortcomings, we propose an alternative visual analytics workflow for multi-parameter ensembles. This workflow builds upon the experiences we have gained when working with researchers from meteorology and computational fluid dynamics, where important structures are often described via specific combinations of multiple physical quantities. It supports users in the selection of meaningful parameter combinations and value ranges of interest, and enables them to

- The authors are with the Computer Graphics & Visualization Group, Technical University of Munich (TUM), 80333 München, Germany. E-mail: {alexander.kumpf, patrick.haertl, westermann}@tum.de, JosefStumpfegger@outlook.de.

Manuscript received 18 Aug. 2020; revised 8 Feb. 2021; accepted 17 Feb. 2021.
Date of publication 24 Feb. 2021; date of current version 2 Sept. 2022.

(Corresponding author: Alexander Kumpf.)

Recommended for acceptance by I. Hotz.

Digital Object Identifier no. 10.1109/TVCG.2021.3061925

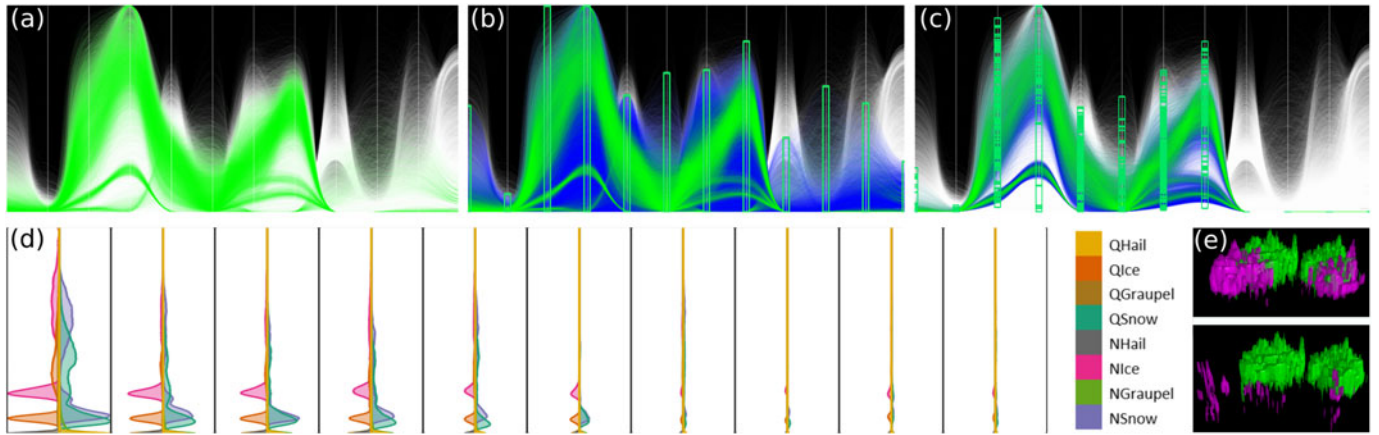


Fig. 1. Method overview: a) A parallel coordinate plot (using polylines for better visibility) of all data points (white polylines) in a representative member of a multi-parameter cloud ensemble, comprising 250K data points with twelve physical parameters per point. Parameter space clustering in this member finds data points shown in green. b) A multi-parameter brush using the cluster's extreme values selects data points which are not in the cluster (blue). c) By using a kD-tree in parameter space, in combination with a principal component representation of the multi-parameter distributions of grouped data points, outliers and data points not following the clustered distribution are discarded. d) An extension of violin plots shows how representative the multi-parameter distribution across the selected cluster is, and how the single parameter distributions differ between the ensemble members. e) Via a multi-parameter 3D view, the spatial distribution of selected data points in different members can be compared and analyzed.

analyse whether similar structures occur in the ensemble and how representative they are.

1.1 Contribution

We introduce a visual analytics solution to investigate the occurrence of multi-parameter distributions in the members of a given simulation ensemble. To obtain an initial distribution that corresponds to a meaningful structure in the data, the system recommends a cluster of data points with similar multi-parameter values and shows them in a Parallel Coordinate Plot (PCP) (Fig. 1a). The system enables users to assess whether the multi-parameter distribution over the selected data points is present in the other ensemble members, and where in the data domain the points contributing to this distribution are located.

To obtain the data points in the other members that follow the multi-parameter distribution of the selected cluster, the cluster's extreme values in each parameter are used as filters. Since this approach selects a hyper-cube in the multi-dimensional parameter space, also regions which do not contain any of the clustered data points are selected (Fig. 1b). When applying the selection to another member with data points in these regions, the multi-parameter distribution over these data points is distorted. In principle, this problem can be addressed by computing a tight hull of the clustered data points in the parameter space, and testing every data point in another ensemble member against this hull. Since this approach is far too computationally expensive in our scenario, we propose an alternative solution building upon an adaptive spatial subdivision scheme. By using a kD-tree over all clustered data points in the parameter space, the selected intervals are automatically split into sub-intervals which enclose the clustered sub-regions more tightly (Fig. 1c). Outlier removal and a covariance analysis of the locations of the data points in each sub-region are then used to focus the analysis on statistically representative trends.

To analyse the similarity of the multi-parameter distributions of selected data points across the ensemble, we

propose a new graphical depiction of the per-member distributions. This depiction is an extension of the classical violin plot [29], which we term Multi-Parameter Violin Plot (MPVP). An MPVP simultaneously shows the distributions of all parameters over a set of data points in a selected member. When designing the MPVPs, we have considered perceptual issues so that users can quickly assess the major differences and similarities between the ensemble members. An MPVP plot showing all members simultaneously is linked to the PCP (Fig. 1d), and it is updated instantly when the user modifies the selection.

A spatial view shows the locations of the selected data points in the 3D domain. We follow the approach by Linsen *et al.* [41] to visualize a tight hull around the brushed data points via isosurface rendering. Since in our use cases the data points live on a voxel grid, surface extraction can be performed efficiently during GPU volume ray-casting. To compare different ensemble members to each other, rays are traced simultaneously against the hull of both the clustered data points and the data points in another member that is picked by the user (Fig. 1e). In this way, the user can quickly compare the spatial extent and mutual overlaps of regions showing a certain multi-parameter distribution. By picking the MPVP of a certain member, the spatial view is instantly updated to show the corresponding region.

The proposed visual analytics workflow builds upon the following specific contributions:

- An automatic refinement of multi-parameter selections using adaptive parameter-space partitioning and multivariate fitting.
- Enhanced violin plots using multiple trace plots and difference-preserving coloring to simultaneously compare multiple parameter distributions.
- A multi-parameter isosurface rendering technique which indicates the parameter intervals that bound a cluster in the spatial domain.

To perform all operations interactively, we provide a flexible and scalable GPU rendering engine for parallel

coordinate plots as well as linked MPVPs and 3D cluster boundary views.

2 RELATED WORK

Our work is related to previous works in the fields of ensemble visualization and multi-parameter visualization.

Ensemble Visualization. Ensemble visualization is related to uncertainty visualization [4], [36], yet it is assumed that the uncertainty is represented by a set of possible data occurrences rather than a stochastic uncertainty model. Previous works in ensemble visualization have addressed especially the question of how to visually convey the ensemble spread of certain physical fields, using either feature- or location-based approaches [50]. For a recent overview of ensemble visualization techniques let us refer to the survey by Wang *et al.* [64]. Feature-based approaches, such as finding visual abstractions of the major trends in ensembles of line or surface features [14], [19], [52], [58], [68], are not subject of our study.

Related to our approach are in particular approaches that represent the spread in scalar-valued ensembles via visual representations of statistical summaries [44], [53]. Others model the variation of field variables by distributions, to reveal major trends and represent the ensemble in a compact way. Single-parameter ensembles are modeled via mixtures of probability density functions to compactly represent their spatial and temporal spread [17], [34], [42], [65]. Hazarika *et al.* [25] propose a copula-based framework for ensemble visualization, in particular to model the statistical dependencies between the scalar values at different locations. This approach was later extended to multi-parameter data [26]. Thompson *et al.* [61] introduce so-called hixels, a meta-representation that encodes a histogram of scalar values in a certain spatial region. Multi-charts build upon a linearization of spatial locations and use side-by-side bar diagrams to compare the ensemble variability at different locations [13]. Höllt *et al.* [30] and later He *et al.* [27] use violin plots [29] to perform location-wise visualization of scalar time-series data. All these approaches visualize single-parameter ensembles, and cannot be extended in a straightforward way to multi-parameter ensembles. We extend these works by first selecting a set of data points with similar multi-parameter values in a reference member, and then determining and visualizing corresponding sets in all other ensemble members.

Multi-Parameter Visualization. At the transition between ensemble visualization and multi-parameter visualization, a number of approaches have been developed to investigate the relationships between multiple input parameters and a single output parameter of a simulation [2], [5], [59], [62]. In contrast to our approach, which aims at a visual comparison of large data sets with many output parameters per data point, these approaches shed light on the sensitivity of simulation results to variations in the input configurations. Thus, at the core of these approaches is a visual analysis of the relationships between input configurations and simulation outputs, rather than the visualization of a single multi-parameter data set.

For the visualization of multi-parameter data, a number of different techniques exist, such as radar charts over pair-

wise scatterplots and correlation heatmaps [38], [43]. Most of these techniques, however, even though effective for rather moderate amounts of data points, do not scale well in the number of points and are, thus, problematic in our scenario. Another technique to directly visualize multi-parameter data points is Parallel Coordinate Plots (PCPs). Popularized by Inselberg [32], [33], a multitude of methods have been proposed to improve the visibility of single data points in PCPs and reduce clutter. Many of these task-specific adaptations are surveyed in the summary of Heinrich and Weiskopf [28]. Johansson and Forsell [35] give an overview of user-centered evaluations of parallel coordinates, Dasgupta *et al.* [12] focus on the use of PCPs to convey uncertainty in the data. For an overview of techniques using multiple coordinated linked views including PCPs let us refer to the summary report by Roberts *et al.* [55].

In our proposed workflow, the user interactively selects lines passing through certain parameter ranges via brushing in the PCP of all data points. Ward [66] introduce n-dimensional axis brushing, where the brushes are created manually using sliders. Fua *et al.* [21], [22] enable viewing the data at multiple resolutions by means of hierarchical clustering. They propose structure-based brushing to select data lying on a wedge in the hierarchical space. Proximity-based coloring, zooming and fading of non-selected nodes in the hierarchy further improves the analysis. Roberts *et al.* [56] propose a brushing technique where a line-strip defines the centers of brushed intervals in each parameter dimension, and the width of selected intervals can be controlled interactively. Brush patterns can be translated vertically and across axes to reveal similar patterns in different parameter ranges. Multiple brushes can be combined and angular histogram glyphs show the value distribution and direction within a brush and axis. We make use of some of the proposed visualization options and integrate additional means into PCPs to automatically refine the set of brushed data points.

Also related to our approach is multi-parameter clustering, which can effectively reduce the number of entities that need to be visualized. Clustering is either performed directly in the multi-dimensional parameter space [41], [47], or via subspace clustering that finds clusters within a sub-set of all parameter dimensions [1], [11], [18], [49]. Non-axis-aligned sub-spaces have been clustered and visualized via dimensionality reduction techniques, i.e., by using projections into linear sub-spaces in which structures in the data points are preserved [45], [48], [69]. For visualizing multi-parameter clusters in the spatial domain, Linsen *et al.* [41] propose isosurface rendering of the cluster boundary via scattered multi-parameter interpolation. Clustering in combination with PCPs has been used by Lex *et al.* [40], by extending PCPs with colored matrices to analyse and compare the quality of cluster assignments using different algorithms. Long and Linsen [63] compute a hierarchy of high density clusters, which are analysed in parallel coordinates and linked views. The bundled parallel coordinates presented by Palmas *et al.* [51] use clustering to improve visual continuity and create links between axis aggregates. Matchmaker [40] introduces curved meta-links between axes. The use of clustering in an ensemble of multi-parameter data sets has been proposed by Kumpf *et al.* [39], by clustering each ensemble member individually and comparing the

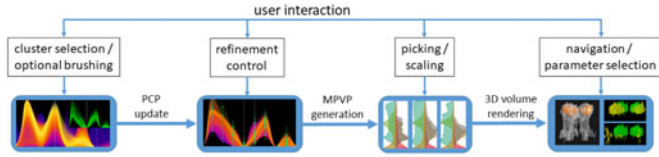


Fig. 2. Workflow overview: To select intervals in multi-parameter space, the user either selects a computed cluster in that space or brushes manually in a PCP. Intervals are refined using adaptive spatial subdivision and non-axis-aligned refinement of sub-intervals, and are applied to all ensemble members. The user controls the refinement via the subdivision depth and the significance level of confidence regions in parameter space. MPVPs are generated to assess the representativeness of a selected multi-parameter distribution across the ensemble. The user can select different drawing styles, and pick a MPVP. Data points belonging to the representative and picked MPVP are rendered in a linked 3D spatial view. The user can navigate in this view and select parameters for which isosurfaces are rendered.

resulting sets of clusters. To do so, however, matchings between clusters in different ensemble members need to be established. Since the number and composition of clusters in each member can change significantly, this becomes prohibitively unfeasible in real-world applications.

3 METHOD OVERVIEW AND DATA

Given an ensemble of 3D multi-parameter fields s_1, \dots, s_n , where each ensemble member s_i is a discrete sampling of a mapping from the continuous spatial domain $\mathbf{S} \subseteq \mathbf{R}^3$ into a K -dimensional parameter space \mathbf{R}^K . The sampling is represented by a finite set of m spatial data points $\mathbf{S}_m = \{x_i : x_i \in \mathbf{S}, i \in \{1, 2, \dots, m\}\} \subset \mathbf{S}$. Without loss of generality, we assume that the data points are given at the vertices of a Cartesian grid, so that tri-linear interpolation can be used for data reconstruction when rendering the 3D spatial view. Fig. 2 provides an overview of the workflow we propose to interactively analyse the occurrence of a selected multi-parameter distribution in all ensemble members.

In a pre-process, the data points in one representative ensemble member are clustered using any suitable clustering algorithm. For instance, HDBSCAN [7] after t-SNE projection of data points is used for Clouds. The representative ensemble member is selected manually by using domain knowledge. For the cloud ensemble, the experts deem the ensemble member that has been simulated with the parameter set closest to the mean of all parameter sets as a suitable representative. When working with ensemble data from the ECMWF Ensemble Prediction System (ENS; e.g., [60]), an unperturbed control forecast (started from the “best” initial conditions) is used as representative member. This control run is available for all ensembles generated by ENS.

A selected cluster in the representative ensemble member is shown in a PCP (see Fig. 1a), and converted to a multi-parameter brush by using the extreme parameter values of all clustered data points. The tool allows switching between curve or line drawing in the PCP. Since for the large number of data points in our application the domain experts found curves to better differentiate the data points, we exclusively use this style throughout the paper. Even though other visualization techniques like scatterplot matrices or multi-charts can be used in this stage, the PCP enables to further refine or enlarge the parameter ranges in a fairly intuitive way by brushing directly on each axis. Following Roberts *et al.* [56],

we use priority rendering of lines w.r.t. a selected value on one of the parameter axes. Therefore, all data points are sorted w.r.t. their distance to the selected parameter value, and blended on top of each other using adjustable opacity in the order of decreasing distance. By further mapping distances to color, the perception of the density and spread of lines passing close to the selected parameter value is enhanced. Since we deal with a single cluster and explicitly emphasize this cluster by coloring the respective points in the PCP, dimension arrangement techniques for PCPs [3] did not lead to any perceptual improvements. Data points not belonging to the selected cluster are then filtered out by an automatic refinement of parameter intervals (Fig. 1c).

To assess the occurrence of the selected multi-parameter distribution in other ensemble members (Fig. 1d), an MPVP is generated for each member. By picking an MPVP, the user selects an ensemble member. This member is compared to the representative member in a 3D view. Both the representative and the selected member are visualized via volumetric ray-casting by rendering a hull with member-specific color around each set of selected data points (Fig. 1e). While the MPVP shows the distribution of parameters over a selected set of data points, the 3D spatial view shows the spatial locations of these points and the geometric shape of the corresponding structures. As proposed by Doleisch *et al.* [15], [16], all views are coordinated and instantly updated when the user performs certain interactions, like brushing in the PCP or picking an MPVP.

The first data set we use is an ensemble of 96 numerical simulations of a growing thunderstorm cloud over a time span of six hours [67], simulated on a $700 \times 500 \times 35$ Cartesian grid with anisotropic spacing. Of this data set “Clouds” we only consider the last time-step. At each data point, twelve precipitation parameters such as hail, water, and rain are given. Parameter vectors are first normalized over the whole ensemble, and data points with a norm less than 0.1 are removed. Roughly 250 k points per ensemble member remain of the initial set of points. The second ensemble is an ECMWF weather forecast of tropical cyclone Karl initialized on 2016-09-22 00:00 UTC with 162 h lead time, in an area from 30N to 80N and 50W to 30E for 30 vertical levels. This leads to roughly 120 k grid locations in a $81 \times 51 \times 30$ grid. Karl is composed of 50 ensemble members and one control forecast simulated with best known initial parameters. Nine parameters such as wind speed, temperature and different precipitation parameters are considered.

4 MULTI-PARAMETER BRUSHING

Brushing parameter ranges in a PCP to select data points corresponding to meaningful structures is challenging. Even though more sophisticated brushing techniques exist, such as area, lasso, or angular brushing [54], we refrain from integrating such techniques into our workflow. This is in particular to relieve the user from specific assumptions about the relationships between different parameters, and let the selection be based solely on value ranges of the available physical quantities. Especially when working with experts from meteorology, we observed that meaningful structures are more or less exclusively defined via parameter intervals.

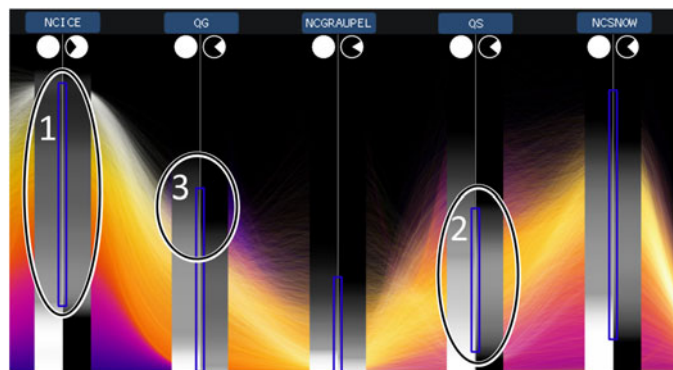


Fig. 3. For an ensemble member, two histograms are drawn for each parameter: One of the values in this member (left) and one of the values in the selected cluster in this member (right). Histograms indicate the representativeness of a parameter interval for a selected cluster, by showing the relation between the frequency of occurrence of parameter values in an ensemble member and a cluster. The ratio between the overall number of selected data points in the ensemble and the cluster is shown by pie-charts above the cluster histograms.

4.1 Cluster-Based Brushing

Our framework supports the user in finding suitable initial parameter intervals, applies these intervals across the ensemble and visualizes for each ensemble member the parameter distribution over the selected data points. This reveals in which ensemble members a certain distribution is present, and how representative this distribution is for a structure in a certain member. The multi-parameter intervals can be refined further via interactive brushing, and are then re-applied to the ensemble.

In many practical applications, and even routinely in weather forecasting, important structures in a data set are often determined via clustering. Even though the results can sometimes be misleading since they depend on intrinsic parameters of the clustering algorithm, extracted clusters often provide a good initial guess about specific relations in the data. Notably, in our application where a cluster consists of a discrete set of data points, it is rather unlikely that identical clusters in different ensemble members exist.

When an initial cluster in a multi-parameter data set is selected, the extreme values of each parameter over all data points in the cluster are used as a multi-dimensional brush in a PCP. Instantly, the selected data points are drawn over all other points in the PCP (Fig. 1a). Furthermore, parameter histograms are embedded into the PCP to compare the number and parameter distribution of data points in the representative member and a selected cluster. We intentionally draw the histograms with shading in gray scale to let empty intervals take the color of the background. We also found that this visual representation enables a more effective comparison of the per-value differences between the two histogram as, for instance, by using rotated conventional histograms using vertical bars.

As seen in Fig. 3, a side-by-side view of parameter histograms – restricted to a selected range – for the initial- and clustered data points enables to quickly reveal two different aspects: First, they indicate how representative a parameter interval is for the selected cluster. For instance, the encircled interval 1 in the parameter range of “number of ice particles” is representative, since most data points where “number of ice particles” is in that interval also belong to

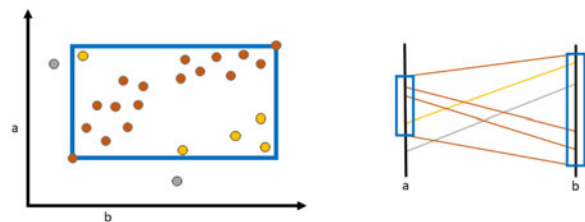


Fig. 4. Left: Some data points (yellow and grey) do not belong to a cluster in 2D parameter space (orange). The cluster's extreme values are indicated by the blue square. Right: The cluster is converted to a brush in a PCP (not all points are drawn). The data point colored yellow is falsely selected by the brush.

the cluster. Interval 2 in the parameter range of snow, on the other hand, is not representative, since for the majority of data points it cannot be decided whether they are in the cluster by only looking at the parameter snow. Note that few outliers in the cluster lead to an enlarged interval 3 for parameter graupel. Via a pie-chart over the histogram of the clustered data points, the ratio between the number of data points in the cluster and in the initial data set is effectively communicated. Second, the cluster histograms reveal whether the distribution of parameter values in the cluster matches that of the data points, or whether the clustering algorithm has determined sub-structures in a certain parameter interval. To provide information about how representative the cluster is and how well the brush fits the cluster, the user can request the ratio of points in a cluster and points selected by the parameter intervals derived from this cluster.

A problem that arises when brushing automatically w.r.t. to the interval ranges of a selected cluster is that the selected set of data points can include points that do not belong to that cluster. Cluster boundaries in parameter space, where the clustering algorithm operates, are usually not axes-aligned, but the intervals define an axis-aligned hyper-cube in the multi-dimensional parameter space. Hence, regions—and data points within them—are selected which are not covered by the cluster (Fig. 4). Another problem arises if the selected parameter intervals are applied to select data points in another ensemble member. If this member has data points in a falsely selected region, the multi-parameter distribution over the selected data points in this member is distorted.

An example demonstrating the mismatch between clustered data points and data points selected by the cluster's extreme values is shown in Fig. 5, using the representative member in the ECMWF data set. The cluster (shown in green) is converted to a multi-parameter brush, which selects the data points rendered in blue. All remaining data points are rendered in white. The cluster is rendered last, so that blue lines shining through in the PCP hint to data points in the selected parameter intervals but not in the cluster.

4.2 Brush Refinement

The selection of data points using the extremal parameter values of a given cluster is sensitive to outliers and can include many more points than contained in the cluster. Furthermore, multiple disconnected sub-structures in parameter space can lead to large parameter ranges being selected. To achieve a tighter fit of the parameter intervals

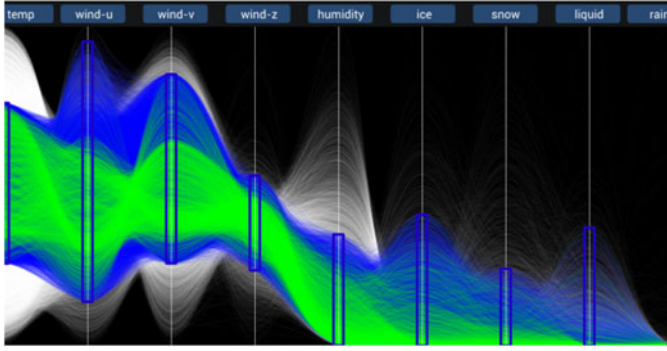


Fig. 5. Illustration of cluster-based brushing. (a) Green lines show a cluster of data points. Blue lines show the data points in that member that are selected by using the cluster's extreme values as a brush. Points of the whole data set are rendered in white.

to the cluster, intervals are refined and re-applied to the data points. The effect of refinement is then also shown by pie-charts over the parameter histograms (Fig. 3), as the pie-chart of the brush and cluster become more similar.

For refinement, a kD-tree is constructed over the data points in the brushed D-dimensional parameter space. Thus, the initially selected hyper-cube is split into smaller cubes separating outliers and contiguous sub-structures. To build the kD-tree, we use the Surface Area Heuristic (SAH) [46] to determine the locations of split hyper-planes along the parameter axes, in combination with a bounding interval-based approach to avoid the construction of very deep kD-trees. When splitting a certain parameter interval, the possible split locations are restricted to the parameter values of the data points in this interval. Therefore, all data points are first sorted w.r.t. the value of this parameter. Then, for all possible split locations the SAH measure C is computed as

$$C = N_l \cdot L_l/L + N_r \cdot L_r/L. \quad (1)$$

Here, N_l and N_r , respectively, are the number of points to the left and right of the current point (with the split point assigned to the sub-interval with more points or shorter length), L is the length of the currently selected parameter interval, and L_l and L_r , respectively, are the lengths of the left and right interval that are generated due to a split. Of all possibilities, the split point for which C is minimized is selected, favoring large sub-intervals with low number of data points. Thus, outliers, contiguous subgroups and

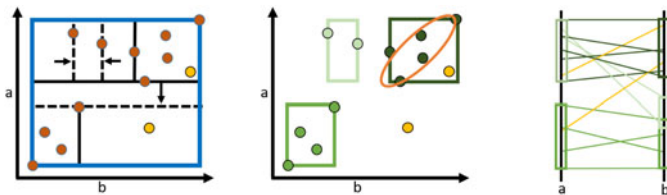


Fig. 6. Left: Parameter intervals derived from a cluster's (orange points) extreme values. First horizontal and then vertical kD-tree-based refinement is indicated by black lines. Dashed lines indicate narrowing of sub-intervals. Vertical refinement works on already narrowed sub-intervals. Yellow points do not belong to the cluster but lie in the derived brush. Middle: Resulting sub-intervals and covariance ellipse of selected point set. Right: Refined brush in PCP. Yellow points are not contained in any sub-interval or covariance ellipse and can be filtered out.

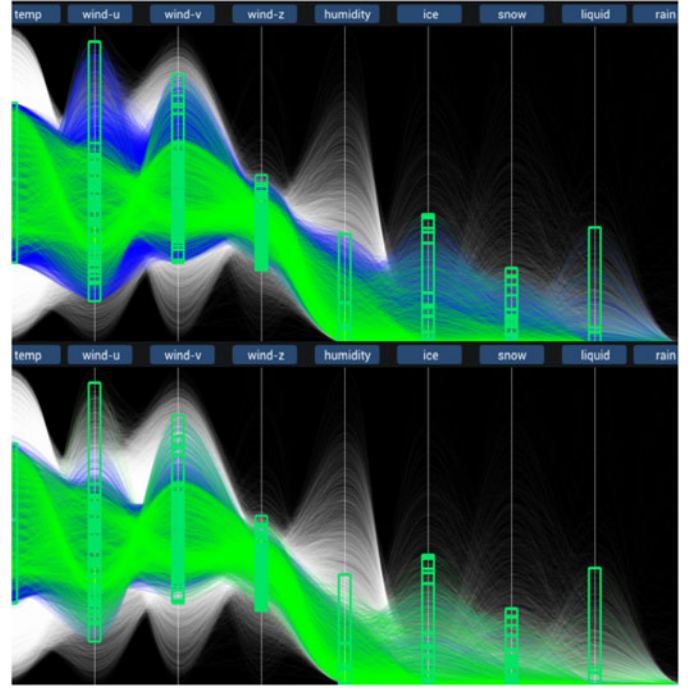


Fig. 7. Same as Fig. 5, but intervals derived from the cluster's extreme values are first refined using the kD-tree-based interval refinement (top), followed by an additional distribution-based rejection of data points using a per-interval confidence region in principal component space (bottom).

empty regions are effectively separated (Fig. 6). Due to alternating splitting along all parameter axes, the refinement considers the structure of data points in the multi-dimensional parameter space.

When using the SAH heuristic, however, a high refinement depth is required until empty regions are effectively separated. In Fig. 6, for instance, only when refining along the horizontal axis for the second time (after all parameter intervals have been refined once), the empty space between the isolated point on the right and the cluster of data points to its left will be separated. To avoid this, after every split the interior boundary of the interval that does not contain the selected split point is refined, so that the resulting interval still bounds all points but is as small as possible (black boundaries in Fig. 6). Thus, already after 1-2 refinements along every parameter axes a good fit is achieved.

Each refined hyper-cube is treated as an independent brush in parameter space. As shown in Fig. 7a, the proposed refinement can effectively prune outliers as well as empty space between them and the clustered structures. On the other hand, the distribution of data points in each hyper-cube can still be such that the derived interval bounds do not faithfully represent this distribution. For instance, the distribution of the data points in the upper right part in Fig. 6 shows a clear diagonal preference, which is not well represented by axis-aligned intervals.

To address this misalignment, we compute additional bounding representations that better fit the distribution of the data points in each hyper-cube. For this, we make use of a principal component analysis of the covariance matrix—containing the covariances between all parameter pairs of the data points—to determine a σ -confidence region in parameter space. When a data point is tested for

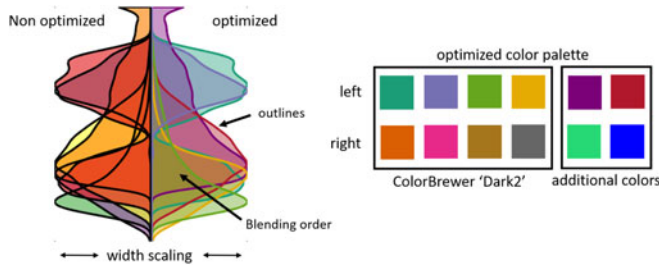


Fig. 8. Design parameters for MPVPs. Right of violin axis: Our favored design and colors. Left of violin axis: An alternative design that is less meaningful and requires more time for interpretation.

containment in a given multi-parameter brush, it is now tested for containment in the confidence ellipses representing the structure of points in these cubes. In this process we pay special attention to degenerated intervals and extremely small eigenvalues, and let corresponding dimensions solely be represented by the unit interval on the principal component axis.

As shown in Fig. 7b, the distribution-based refinement strategy prunes a number of additional data points that are not aligned with the majority of data points in a hyper-cube. This is also advantageous when applying the selected multi-parameter interval to other ensemble members, since it enables to consider the structure of data points in the parameter space rather than solely their locations.

5 ENSEMBLE ANALYSIS

Once a suitable multi-parameter brush has been obtained from the cluster's extreme values and further refined by using the kD-tree and covariance analysis, this brush is applied to all other ensemble members. To reveal how representative the multi-parameter distribution of the selected points is for the ensemble, i.e., whether a significantly large set of data points with similar distributions exists in each member, we show the distributions for each member side-by-side. Juxtaposition of the single member visualizations can reduce cluttering, yet, on the other hand, the context between similar features in different members can get lost [24]. While this is certainly a limitation of our approach, we try to reduce this limitation by a number of design decisions concerning the shape, color and ordering of the graphical elements used.

5.1 Multi-Parameter Violin Plots

The MPVPs we propose to visualize a multi-parameter distribution over a set of data points in a single member build upon density trace plots [10], which were later modified by Hintze and Nelson [29] towards so-called violin plots. A violin plot is a vertical or horizontal density plot with the probability density curve of a parameter on both sides of a vertical axis, optionally accompanied by a box plot of the parameter values in its interior. The book by Chambers [9] gives a thorough overview of the different variants of this type of plot. Höllt *et al.* [31] show the density curves of 2 different parameters on either side of the vertical axis to enable the comparison of 2 scalar value distributions. In a violin plot, multi-modal data distributions appear as multiple peaks in the density curve, and the distance between the

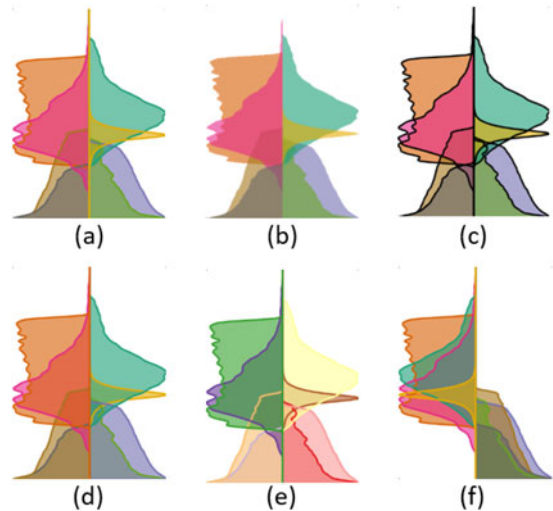


Fig. 9. Different visual designs of MPVPs. a) Our proposed design with bold opaque outlines in the region color, as well as perception-aware color selection, left-right assignment of parameters, and drawing order. b) No outlines. c) Black outlines. d) Random drawing order and e) random colors of single violin plots. f) Random left-right assignment of parameter curves.

curve from the axis provides information about the number of data points contributing to a certain parameter value. The area under each density curve is either colored uniquely, or different colors are used in one plot to distinguish between multiple layers in the data.

The extension we propose to make a violin plot applicable to a multi-parameter distribution is to overlay multiple violin plots—each with its own colors on either side of the axis—in a single MPVP. An illustration of the parameters we have considered in the design of MPVPs, with an ad-hoc selection on the left and our selected design parameters on the right, is shown in Fig. 8. In this work, we have focused on the use of MPVPs to visualize twelve parameters simultaneously. Designs for other numbers of parameters can be derived as discussed in the following, yet for larger numbers the increasing chance of using similar colors for different parameters will limit scalability.

The first design parameter we found important is the style and color of the outlines of the regions covered by the violins, i.e., the density curves. We evaluated 3 different possibilities (Fig. 9): a) A bold outline in the color of the region. b) No outline. c) A black outline. Drawing no outline has 2 apparent drawbacks compared to the other options: First, regions colored with a rather light color do not stand out against the background. Second, when regions overlap and the resulting color is similar to the color of the last rendered violin, the shape of the covered region can no longer be perceived clearly. Drawing a black outline can effectively reveal the shape of each region—also when blended with some other regions—yet since all regions have the same outline color it becomes difficult to distinguish between them where many contours appear. The first option (Fig. 9a) turned out to be most effective, standing out against the background and conveying the single plots in the overlap regions.

When multiple violins overlap, their colors need to be blended. Overwriting colors by the color of the last drawn violin can occlude violins that are completely covered by this last one. One can also select a color for an overlap

region that depends, for instance, on how many violins overlap in this region. This, however, makes it impossible to grasp the connection between an overlap region and the violins that contribute to it, since the violins' colors are not preserved in the color of the overlap region. In addition, this coloring results in abrupt color changes between the regions that occur due to different numbers of overlapping violins. Another issue that needs to be considered is that the visual context between multiple MPVPs should not be lost, meaning that overlap regions with the same set of contributing violins should be easily perceivable across a set of MPVPs shown side-by-side.

To address these issues, we use α -blending when drawing the single violins, and explicitly enforce a certain drawing order (Fig. 9a). The violins are drawn in the order of decreasing area between the density curve and the violin axis, so that the chance to completely over-blend a single violin becomes lower, even though still possible. The drawing order is established from the representative ensemble member and used for the visualization of all other members. Violin plots are then blended atop of each other, i.e., we start with a white background $C = (1, 1, 1)$ and then update this color via $C = \alpha \cdot C_V + (1 - \alpha) \cdot C$ whenever drawing a violin with color C_V and opacity α . An α value of 0.4 is used to let overlaid colors shine through even when more than two layers are present. Fig. 9d shows that some regions get lost when rendering and blending is performed in an arbitrary order.

To avoid hardly distinguishable blend colors, we follow the recommendation of Gama and Goncalves [23]. As all complementary colors (being opposite on the color wheel) blend into a similar color, humans can better distinguish the blend colors of different pairs when these pairs are close on the color wheel. Since we have up to twelve parameters and, thus, need up to six colors on each side of the violin axis, we select two sets of six colors each. The eight colors of the ColorBrewer palette Dark2 are sorted according to their hue, and split into two groups of four (right and left violins) such that the maximal distance between hues is minimal in each group (Fig. 8). The additional two colors per side are chosen from the other side of the color wheel to avoid too many similar colors, yet with the trade-off of few complementary colors. Figs. 8 (right) and 9a are both generated with these colors. In contrast, and as seen in Figs. 8 (left) and 9e, when using an arbitrary color selection, either very similar colors for different violins or pairs of complementary colors with the same blend color can be selected.

We further determine an assignment of ensemble parameters to the left or right of the violin axis that reduces the mutual overlaps between the single violin plots on either side. Therefore, a similarity matrix $D = \{d_{i,j}\}$ is computed, which represents the pair-wise overlaps of two parameter regions i and j . Values a_{ij} indicate the percentage of the region of i that is covered by j . The minimum $d_{i,j} = \min(a_{ij}, a_{ji})$ defines the similarity between the violin plots of parameters i and j . Large mutual overlap indicates overall similar shape of the plots, implying that the plots should be drawn on different sides of the violin axis. Small overlap, on the other hand, is not critical, since due to the drawing order the smaller plots are drawn atop of the larger ones. Iteratively, the pair of parameters with highest similarity

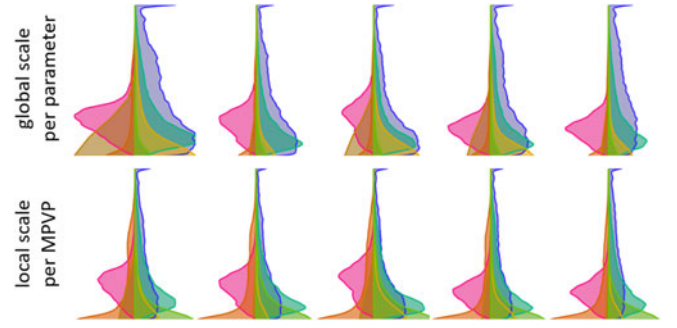


Fig. 10. MPVPs of 5 ensemble members. Top: For each parameter, the horizontal scale, i.e., the value at the left and right border, is set to the maximum value of the curves for this parameter over all ensemble members. Bottom: For each MPVP, the horizontal scale is set to the maximum value of all parameter curves of the corresponding ensemble.

value $d_{i,j}$ is determined, and these parameters are assigned to opposite axis sides. To decide which of the two parameters is placed left and which right, the violins already placed left k_l and right k_r are considered. In particular, the assignment is selected that minimizes the sum of the maximum similarities between the parameter pairs on the left and right side, i.e.,

$$\arg \min_{\rho \in \{(r,l), (l,r)\}} \left(\max_{k_{\rho[0]}} d_{i,k_{\rho[0]}} + \max_{k_{\rho[1]}} d_{j,k_{\rho[1]}} \right).$$

Here, r and l indicate the right and left side, and $k_{\rho[i]}$ indicates already assigned parameters on either side. As can be seen in Fig. 8f, with no explicit assignment of parameter curves to the sides of the violin axis, large overlap regions can occur and absorb whole single violin plots.

Another design decision is how to scale the single violin plots in the horizontal. Here, we provide two different scaling modes: The first one uses one scale for each parameter, and applies these scales to the respective parameters of every MPVP. For a particular parameter, the largest density value in any of the MPVPs is computed, and this value is set as the interval bound for that parameter in every MPVP (Fig. 10 (top)). Even though this mode distorts the proportions between the parameters in one single MPVP, it is necessary when the total densities of parameters differ. It is in particular useful when showing multiple MPVPs side-by-side and comparing the parameter distributions across the ensemble. The second mode lets every single MPVP use its own scale, which is applied to all parameter curves in this plot. In this way, the available drawing space in each plot can be used as good as possible, without sacrificing the relative proportions between the parameters (Fig. 10 (bottom)). This second mode is used when a fine-granular analysis of the multi-parameter distribution in a single member is desired.

5.2 Spatial Multi-Parameter View

To further support the user in examining the spatial locations of selected data points, a 3D view is linked to both the PCP and MPVP chart. Per default, the 3D view is synchronized with the PCP, so that any selection of a representative member or changes of the multi-parameter brush due to the selection of a cluster, refinement, or user interaction

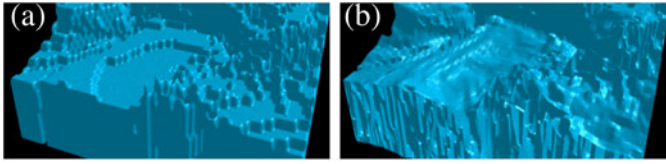


Fig. 11. a) Iso-surface in a binary field derived from a multi-parameter field of the representative member in Karl. b) Same iso-surface computed via smooth interpolation in each parameter field.

immediately triggers an update of the spatial view. In the spatial view, the locations of all data points in the selected member and the points in the refined cluster in that member are instantly rendered (Fig. 11). The user can also select an MPVP, in which case the 3D view is linked to the ensemble member shown in this MPVP. Then, instead of all data points in the representative member, the data points in the corresponding member that are selected via the refined multi-parameter brush are shown in green (e.g., top right in Fig. 17).

Rendering the selected data points in 3D space is performed in two different ways (Fig. 11), depending on whether a high performance or high quality visualization is favoured. In either case, surfaces enclosing the data points are rendered semi-transparently, by using adjustable opacity values.

Binary Representation. During interaction, for instance, if the user manually changes the multi-parameter brush, selected data points are converted into binary values and rendered via isosurface ray-casting in the resulting binary field. A multi-parameter value is set to one, if all parameters are within the selected parameter intervals, otherwise the value is set to zero. In the resulting binary volume, ray-casting is used against the iso-contour to the value of 0.1 using tri-linear interpolation. Optionally, the binary value distribution can be smoothed via an adaptive Gaussian low-pass filter, which can keep values of one unchanged to avoid removing small isolated data points. Since the computation of a binary volume is extremely fast on the GPU, even for Clouds with a size of $700 \times 60 \times 500$, fully interactive frame rates are achieved. Multiple binary fields can be stored and visualized simultaneously. We use this option to show the locations of brushed data points in the representative member relative to locations of all data points in that member (Fig. 12a), and to compare the locations of selected data points in different members (top right in Figs. 17 and 18).

Continuous Field Representation. To generate a high quality visualization, ray-casting is performed simultaneously in all single parameter fields. During ray-casting it is checked at every sample point whether all interpolated parameter values are within the selected parameter intervals. The transition into and out of these regions is detected by a change from “at least one parameter out” to “all parameters in”, and vice versa. For shading, the gradient in the field of the parameter that was last going “in” (when entering the region) and first going “out” (when exiting the region) is used. Computationally, this method is far more complex than the first rendering option due to the significantly higher texture accuracy and number of texture look-up operations during ray-casting. On the other hand, it considers the smooth variation of physical parameters across the domain and leads to a continuous multi-field visualization.

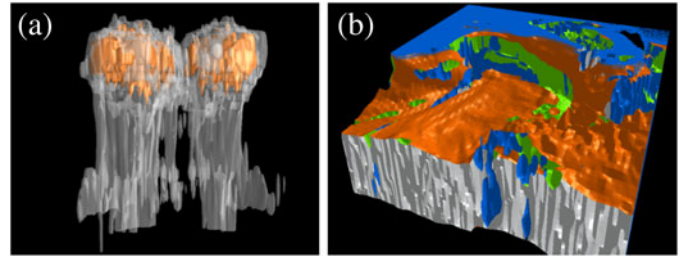


Fig. 12. a) Surfaces enclose all (grey) and brushed (orange) data points in the representative ensemble member of Clouds. b) Continuous multi-parameter surface rendering in Karl shows which parameter intervals limit the surface structure in 3D space. Wind-speed in white, temperature in orange, humidity in green and precipitation quantities in blue.

[20]. Furthermore, it can effectively show which single-parameter brush bounds the rendered surface, by coloring surface points according to which parameter triggered the surface hit (Fig. 12b).

6 GPU IMPLEMENTATION

To enable interactive brushing and applying the brush instantly to all ensemble members, we have implemented a high-performance implementation on the GPU. It is particularly tailored to efficiently handle large multi-parameter ensembles, to be able to instantly update PCPs, MPVPs and 3D views upon user interaction.

The GPU implementation is index based, so that once the data points with assigned index reside in GPU memory, many operations require touching on far less memory compared to an explicit representation. In order to support efficient and parallel range queries for brushing, a boolean buffer containing the activation of each index is present in each drawable instance. If a brush update occurs, a compute shader processes all indices of the drawable instance in parallel and updates the activation of each index. Afterwards a second compute shader then uses the so computed activations to update the indexbuffer for rendering. Furthermore, histogram-based similarity queries between different ensemble members or between members and a given histogram can be performed in a highly efficient way. All data points can be processed in parallel, and by using atomic memory operations on the GPU, parallel increment operations on the values in the GPU histogram buffer can be issued in an exclusive way.

Our implementation via the Vulkan graphics API (published on <https://github.com/wavestoweather/PCViewer>) enables cross-platform support and high rendering performance. On a standard desktop PC—an Intel Xeon E5-1650 v2 CPU with 6×3.50 GHz and NVIDIA GeForce GTX 1070 graphics card with eight GB VRAM—the performance is high enough to interactively analyse the set of ensemble members via brushing, MPVP visualization, and volume rendering, i.e., 2.5 million data points with twelve parameters each in Clouds. It takes roughly 0.5 seconds to execute a normal brush operation (two seconds for a multivariate refined kD-brush) and render the result for ten ensemble members simultaneously, including histograms, and 200 milliseconds for a single member. Volume rendering using the binary and continuous representation takes roughly 0.1 seconds and six seconds for Clouds.

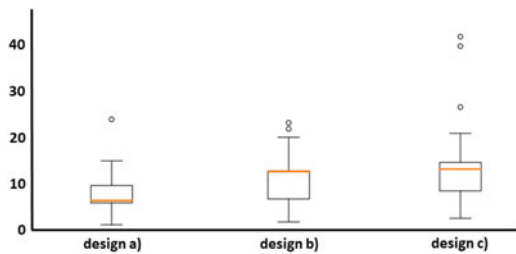


Fig. 13. User times in seconds to locate and classify parameters in a MPVP.

7 EVALUATION

7.1 MPVP Evaluation

The effectiveness of different MPVP designs for conveying multi-parameter distributions was studied in two informal user studies. In a first study, the user experience when using the six different visual designs shown in Fig. 9 was studied. A second study was conducted to evaluate how well MPVPs can reveal the (dis-)similarities between multi-parameter distributions. For each study we recruited five computer science students, all having a background in computer graphics and visualization. No student took part in both studies. The participants were selected to have no color vision deficiency. The students were exposed to MPVPs for the first time, and none of them knew the visualized ensemble fields. Some MPVPs not included in the evaluation were shown and explained to the users beforehand. All MPVPs were shown with global scaling in use.

In the first study, for each design option we created ten MPVPs using randomly selected multi-parameter distributions from Clouds and Karl. In each MPVP, we randomly selected a parameter and classified it into {left, right}, according to the side of the MPVP axis it is shown, and {peak, multi-peak, plain}, according to whether the distribution shows a single or multiple clearly pronounced peaks or a more plain distribution. We selected parameters which could be classified with high certainty. In a web interface, the MPVPs were shown consecutively, and the users were asked to classify the selected parameters as fast as possible and push a button to start the next evaluation. The selected parameters were indicated by colors and each question appeared for all six design options in random order. The time required for classifying each MPVP was measured and the average time was computed for each user and the six different design options. Finally, for each option the average time and standard deviation (both in seconds) over all users were computed.

While no mis-classification was recorded when using the designs a), b) and c), not all parameters could be identified correctly with designs d), e) and f). This is due to occlusions

that are produced by these designs, resulting in parameters covered entirely or to a major extent. Therefore, we excluded designs d), e), and f) from the study. For the options a), b) and c), the recorded measures are represented via box plots in Fig. 13.

It can be seen that designs b) and c) are significantly less effective than design a). When asking the users about their subjective impression, it turns out that designs b) and c) require users to spend more time on regions where many single-parameter distributions overlap. In these regions, design c) leads to increasing visual clutter due to the crossing of many indistinguishable outlines. Since differences in the colors of the single-parameter distributions are washed out due to blending in the overlap regions, design b) makes it difficult to associate perceived colors to the initial colors of the parameters. Due to the drawing of outlines in the unperturbed colors of the single parameters, a more effective shape perception was reported when using design a).

In the second study, our goal was to analyse how well MPVPs can be ordered by a user w.r.t. increasing distance to an initial multi-parameter distribution, and how well relative differences between the single-parameter distributions can be assessed. To perform this evaluation, we have created a synthetic ensemble comprising 20 members which increasingly differ from a representative member (Fig. 14). The representative member consists of 10 k data points, each with twelve normally distributed artificial parameters. The means and standard deviations of the normal distributions were selected randomly. For the i th ensemble member, up to i of the initial distributions are randomly perturbed. The means are changed by up to 0.3 of the initial mean, and the initial standard deviation is scaled randomly by factors between 0.4 to 2.5. In this way, the ensemble members tend to become increasingly different to the initial reference.

We asked the users (T1) to order the MPVPs w.r.t. decreasing similarity to the representative member, and (T2) to classify each MPVP and parameter according to how much they differ (in categories of low, medium, high) from the representative one. Via T1, our goal was to study the effectiveness of MPVPs to support a visually-guided comparison of many multi-parameter distributions, which, in contrast to a numerical similarity score, enables to consider the mutual relationships between the shapes of the single-parameter distributions in one MPVP and across them. T2 serves as a hint on how well certain parameters can be perceived in the MPVP and whether the proposed visual design leads to similar results for all parameters. In T3, we asked to identify the four to six most dissimilar members solely based on the first (green on the left) and fourth parameter (purple on the right), to shed light on the perception of distribution patterns involving multiple parameters.

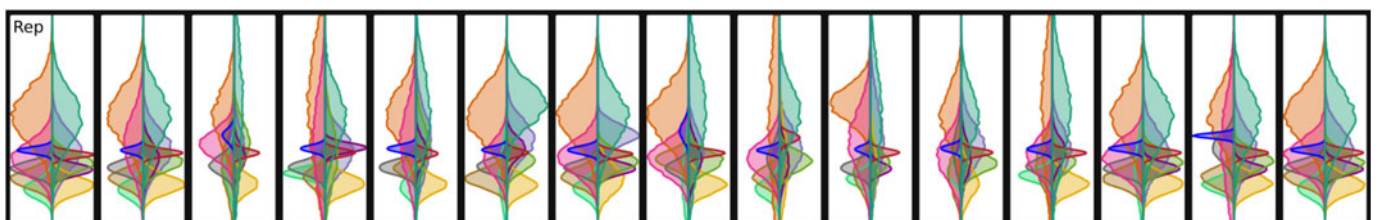


Fig. 14. MPVPs of the synthetic data set in the random order they were presented to the participants.

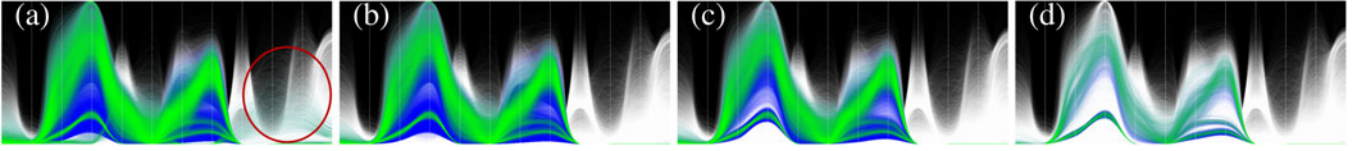


Fig. 15. a) Data points in Clouds selected by a cluster's extreme values and reduced about empty intervals after 20 kD-tree refinement steps (all data points in the representative member in white, data points remaining from the initial cluster in green, and from the extreme value brush in blue). Many outliers remain (encircled). b) kD-tree partitions with less than seven points are removed. c) and d) In each partition, data points not within 1.25 and 0.75 standard deviation from the mean, respectively, are removed.

The results of task T1 were compared to an automatic ordering using the chi-squared similarity measure [57]

$$\chi^2(h_{rep}, h_b) = \frac{1}{2} \sum_i \frac{(h_{rep,i} - h_{b,i})^2}{h_{rep,i} + h_{b,i}}, \quad (2)$$

where $h_{rep,i}$ and $h_{b,i}$, respectively, refer to the i th histogram bin in the representative member and the current member to be compared. The measure is computed for every parameter and accumulated. Scaling by the number of points in the representative member leads to a maximal distance of one per parameter. Our system initially shows the MPVPs in the order of decreasing chi-squared measure to the representative member. It is worth noting, however, that this ordering can be misleading since it does not consider the shape of the single parameter distributions. For instance, a MPVP that is identical in shape to the representative but scaled, can produce the same chi-squared measure than a MPVP where half of the distributions match exactly but the other half is completely different. Since the number of selected data points per ensemble member varies, especially low chi-squared measures require further inspection as provided by MPVPs.

When a user places an MPVP at the wrong position in the ordered sequence, the distance to the correct position is added to an overall error. The resulting error was then also compared to the error that is introduced when 1,000 random permutations of the MPVPs are used. The participants were in average off by 2.1 positions for each MPVP, and 4.7 can be expected by random permutations. Even though 2.1 might seem high at first, it has to be considered that some sub-sets of MPVPs only show very subtle changes, and their positions are, thus, interchangeable. This experiment indicates that the proposed visual design of MPVPs facilitates an effective judgment of the similarity of multi-parameter distributions regarding the shape, interval, and scale of the single-parameter distributions.

For T2, we computed for each MPVP and every parameter the similarity to the corresponding parameter in the representative member via the chi-squared similarity measure. Parameters were then classified according to categories determined by the users, and discrepancies between categories were accumulated over all members. Overall, the strongest mis-classifications were spread across all parameters, even for those parameters mainly concentrated in the middle part of the MPVP axis where significant overdraw is perceived. Surprisingly, the two parameters occupying the largest area (orange on the left and green on the right) did not perform best. Furthermore, user feedback revealed that some users put more attention on the shape of the distributions, while others were more concerned about the exact

vertical position. Both assessment strategies are provided by MPVPs.

Finally, users were asked to judge the similarity of ensemble members based on two selected parameters. Three users identified almost exactly the members that were indicated by the χ^2 -metric. Only up to two members were wrongly classified, both with very small distance values. The other users were correct in the most similar members, but judged the most severe changes differently. In the concluding discussion, the selections had to be justified and it turned out that most participants focused on the shape of the single parameter distributions relative to each other, information that is difficult to encode in a numerical score. As there is no gold standard measure that can reliably compare distributions regarding different use cases, a visual analysis seems inevitable. As even the optimized rendering cannot always avoid over-blending of parameter curves, certain parameters can be excluded from the rendering. This option, however, was disabled in the user study.

7.2 Parameter Space Refinement

To evaluate the effects of parameter space refinement, we use PCPs in combination with MPVPs. In Fig. 15, four PCPs show different subsets of the data points in the representative member of Clouds. The data points in the initial cluster and selected by the cluster's extreme parameter values (min-max brush) are shown in Figs. 1a and 1b. Fig. 15a shows the data points when selecting the intervals that result from 20 (up to three along each parameter axis) kD-tree refinements and removing empty intervals. While some data points not in the cluster are removed, many outliers remain (see encircled data points). Next, partitions with less than eleven points (less points than dimensions) (Fig. 15b) and points not within 1.25 standard deviation from the mean of points in a partition (Fig. 15c) are removed. The standard deviation is then further decreased to 1 (Fig. 1c) and 0.7 (Fig. 15d), to demonstrate the corresponding removal of data points. The resulting PCPs indicate an increasingly tight enclosing of the data points in the initially selected cluster, yet it is difficult to judge when the selected multi-parameter filter starts removing data points too aggressively.

Fig. 16 shows the MPVPs (using global and local scaling) corresponding to the initially clustered data points and the selections in Figs. 1 and 15. By using global scaling, the reduction of data points can be perceived clearly, but some single parameter distributions cannot be distinguished any more. In particular, since there are significantly more data points that are selected by the cluster's extreme values than in the cluster, the multi-parameter distribution across the cluster can hardly be seen. On the other hand, one can see

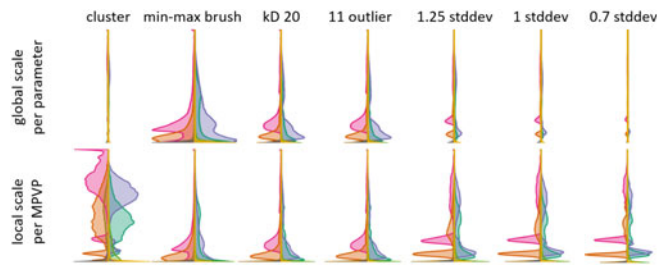


Fig. 16. MPVPs corresponding to the multi-parameter distributions of the clustered and successively refined data points.

on which parameters and in which interval the applied refinements have the largest effect, enabling to focus on structures that persist over many refinement levels.

By using MPVPs with local scale, the changes – also relative to each other – of the single-parameter distributions can be assessed effectively. In particular, Fig. 16 shows two persistent peaks on the left side of the MPVP axis (brown and pink) which are successively separated by the refinements and converge towards the distribution across the cluster in this parameter range. The same holds for the two smaller peaks on the right side of the axis (green and violet), eventually showing good agreement with the multi-parameter distribution across the cluster. Due to the local scaling, however, the distributions over parameter intervals with less points become squeezed and cannot be assessed very well (upper half of the MPVP axis). In the shown example, even though the distributions over these intervals fit the distributions in the cluster very well, this cannot be well perceived from the MPVP due to scaling w.r.t. the size of the largest peak. In such a case, the user can further scale the MPVP manually or with logarithmic bin scaling, so that also the distribution of squeezed distributions can be well perceived and compared to the cluster.

The shown example demonstrates that a tight covering of the clustered data points in the multi-parameter space can be achieved, but it also hints at some limitations of our approach. Since the domain is partitioned using a kD-tree-based refinement with automatic selection of the partition boundaries, it is difficult to determine the depth of the refinement so that not too many but also not too few data points are removed. It can happen, in particular, that data points belonging to a coherent structure in the parameter space are separated. Then, further refinements can even partition into fewer and fewer points that are eventually removed. The same is true for selecting the standard deviation that is used to prune data points within a partition, which might be too small and removes too many points. On the other hand, in combination with MPVPs, which show the multi-parameter distributions of the initially clustered data points and after refinement, the user can interactively select different refinement strengths. For domain experts it is in particular useful to see the correspondence between refinement and pruning to the individual parameter curves, so that meaningful parameter intervals can be selected.

8 APPLICATION ANALYSIS

For the forecast ensemble Karl, Fig. 5 shows in green the cluster that was initially selected in the representative

member. Selection of data points using the cluster's extreme values per parameter give roughly 15 times as many points as in the cluster (blue lines in Fig. 5). Notably, however, MPVPs with local scale (Figs. 17a and 17b) already indicate good agreement between the multi-parameter distribution across the cluster and the selected set of data points. The selection was further refined by removing intervals with less than seven points via a kD-tree. After two kD-tree refinements per dimension, the number of points is reduced to roughly three times the data points in the cluster (Fig. 7 (top)), and covariance-based refinement using a 1.0-confidence ellipse further reduces this number to 1.2 times the initial points (Fig. 7 (bottom)). The PCP seems to indicate an even better fit of the refinement to the initial data points in the cluster, but the corresponding MPVPs (Figs. 17c and 17d) show in fact less correspondence with the MPVP of the initially clustered data points (see in particular the peaks on the left side of the MPVP axis). Thus, it was decided to use kD-tree-based refinement without additional refinements using confidence regions in parameter space. The resulting parameter selection was then applied to all ensemble members. The corresponding MPVPs of 20 members are shown in Fig. 17 (bottom).

The comparison of per-member MPVPs indicates very high representativeness of the selected multi-parameter distribution. This observation is surprising, since the forecasts were simulated using different initial conditions to cover as good as possible the spread in the uncertain weather predictions, and this spread is usually high for high-impact weather events like Karl. In the current case, the comparative analysis using MPVPs indicates very good agreement of all forecast simulations with the representative forecast in which the initial cluster was selected. The MPVPs strongly agree in all single-parameter distributions, both w.r.t. the number of selected data points as well as the shape and location of the distributions. A similar analysis would now have to be performed using other clusters to confirm the agreement in all predicted structures.

Notably, a comparative structure-based analysis using multi-parameter distributions in many ensemble members was not possible until now. Current operational and scientific workflows support the analysis of the ensemble spread via clustering of single-parameter fields, but they do not consider the mutual relationships between the multi-parameter distributions across the ensemble. In particular, our proposed workflow can be used in principle to derive new feature-descriptors, by applying meteorological feature extraction first, and then analysing the multi-parameter distributions across the spatial locations in which the feature is located. Via distribution matching, the sets of data points in other ensemble members with corresponding multi-parameter distribution – possibly encoding the same feature – can be classified. By considering stochastic variations in the distributions, the uncertainty in the found structures could even be quantified.

The 3D spatial view of the representative member in Karl (Fig. 17 (top, right)) indicates that the data points in the cluster concentrate around the region in which the storm is present, i.e., it seems that the cluster has determined the multi-parameter distribution that is obeyed by the locations covered by the storm. By simultaneously visualizing

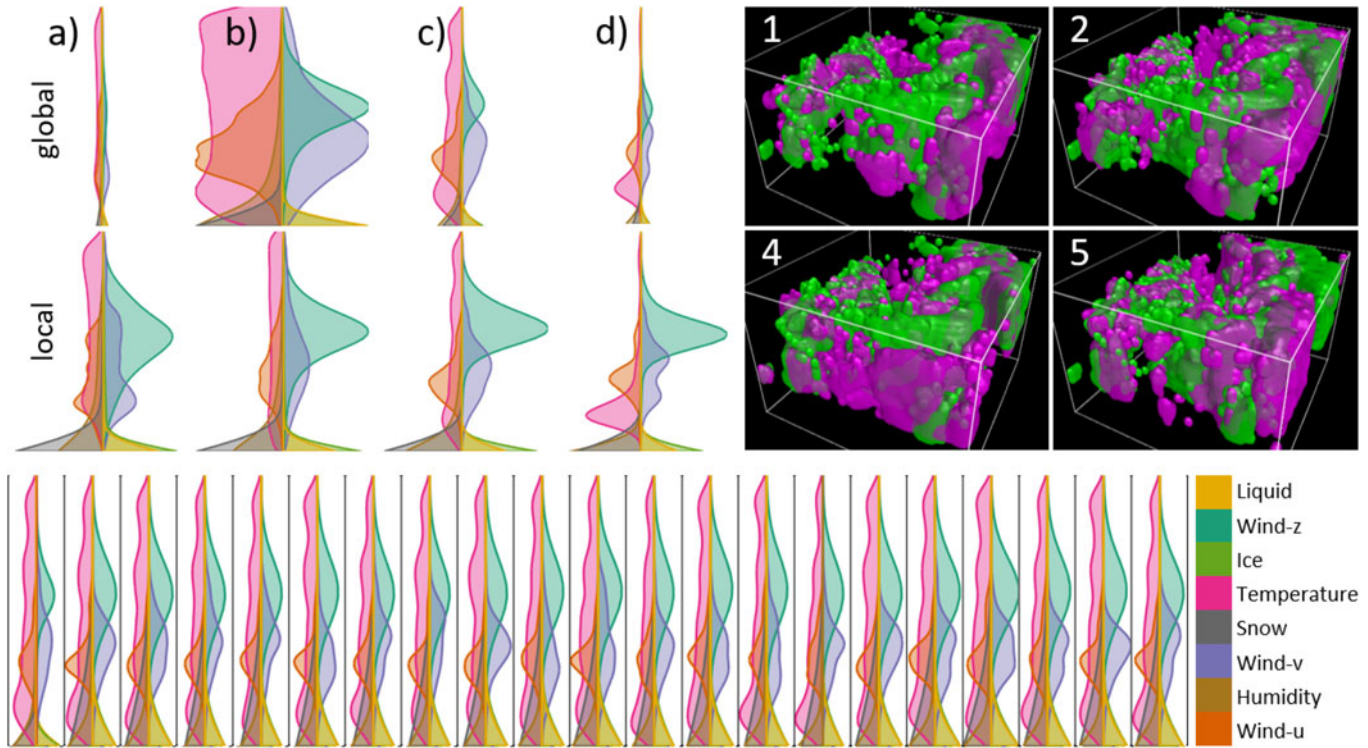


Fig. 17. Cyclone Karl. Top left: In the representative member, MPVPs with global and local scaling show the multi-parameter distribution over all clustered data points (a), over all data points that are selected via the cluster's extreme values (b), and over all data points that remain in the parameter intervals refined via kD-tree (c) and covariance analysis (d). Top right: Locations of brushed points in the representative member (green) and members indicated by numbers (violet). Bottom: MPVPs with global width scaling for the representative and the first 20 ensemble members.

another ensemble member, it can be seen that both ensemble members agree in the region in which selected data points are located and differ only in some smaller structures around that region. This is also confirmed by the MPVPs, which show slightly enlarged parameter distributions, indicating selected data points which do not belong to the cluster. Another interesting finding is w.r.t. to the characteristics of cyclons, which come with fast moving air masses and high precipitation. This reflects in high wind components in the selected data points, but at the majority of selected points both humidity and liquid values are in the lower parameter ranges (brown and yellow distributions). Thus, it can be conjectured that a more fine-granular selection is required – especially regarding the parameters related to humidity – to more faithfully select the storm structure. With the proposed workflow, the domain expert can now perform manual brushing to further refine the selected structure in the parameter space.

For Clouds, the same analysis as for Karl was performed. The analysis of the initially selected cluster using MPVPs reveals two separate structures in the parameter Nlce, shown by the pink peak and wider bulk on the left side of the axis of the MPVP in Fig. 18a (local scaling). The two structures are also shown in the spatial view in Fig. 12.

A selection of data points solely based on the cluster's extreme values (Fig. 1 (top, middle)), Fig. 18b)) cannot separate these structures and includes far too many data points. Figs. 18c and 18d show the effects when refining the initial selection via a kD-tree (20 refinement steps and intervals with less than eleven data points removed) and covariance-based refinement (using a standard deviation of 1.0). Covariance-based refinement leads to a multi-parameter

distribution across the selected points that is very similar to the distribution over the initially clustered data points, and it effectively separates the two cloud structures.

In contrast to Karl, the comparison of 20 members using per-member MPVPs with global scale (Fig. 1) indicates far lower representativeness of the selected multi-parameter distribution in Clouds. This is with respect to the number of selected data points and the parameter distributions across these points. It can immediately be seen that some members do not agree very well with the selected multi-parameter distribution. MPVPs with local scale (Fig. 18 (bottom)) enable to assess the relative occurrences of parameter values per member. This allows for a comparative analysis of the shapes of the single-parameter distributions, effectively indicating common trends and variations.

The analysis reveals the representative multi-parameter distribution also in other members, indicated by peak-like distributions in the MPVPs, yet to a far less pronounced extent regarding the number of contributing data points. The domain experts found it in particular interesting that the two green and violet peaks associated with snow on the right side of the MPVP axis are present in every member, yet with slightly different intensities and vastly different numbers of data points. This seems to indicate a bias of some simulation members, which needs to be further examined by investigating the used simulation parameters and models. Especially ensemble members 11 and 58 differ significantly in the parameters not represented by the peaks. In this case, manual brushing can be used to broaden the parameter intervals, and thus to analyse whether some of the structures in the representative member occur in other members but have shifted in the parameter domain. When

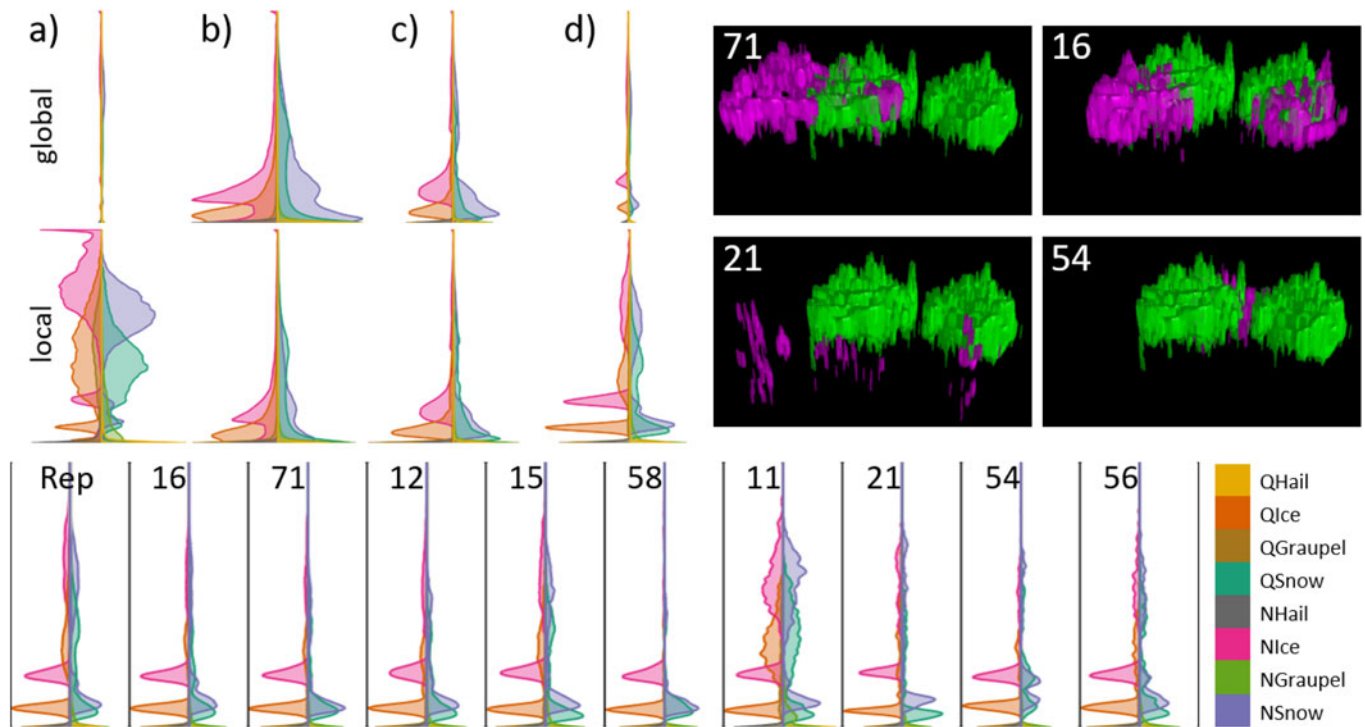


Fig. 18. Clouds. Top left: In the representative member, MPVPs with global and local scaling for data points a) in the cluster, and brushed via b) cluster extreme values, c) kD-tree-refined intervals, d) covariance-refined intervals. Top right: Locations of brushed points in the representative member (green) and members indicated by numbers (violet). Bottom: MPVPs with local width scaling for the same members as in Fig. 1.

visualizing the 3D locations of selected data points in the representative member and the most similar member 16 (Fig. 18 (top, right)), low agreement is revealed. In some of the ensemble members, even though they agree in the shape of the single parameters, the locations and shapes of the selected structures in 3D space differ significantly. This seems to indicate that similar parameter distributions do not, in general, correspond to similar spatial shape of the corresponding meteorological structures.

9 DISCUSSION AND CONCLUSION

We have proposed visualization techniques and a workflow for analysing multi-parameter distributions in the members of an ensemble of 3D fields. To show simultaneously multiple parameter distributions over the data points in a single member, we have proposed an extension of violin plots (MPVPs) for multi-parameter data. By using MPVPs, the representativeness of a selected multi-parameter distribution for the ensemble as well as outliers and major trends in the distributions are efficiently revealed. By a linked spatial view, the locations of selected data points in different members can be visualized and compared to each other.

The limitations of the proposed workflow are regarding a more controllable refinement of initially selected data points and the side-by-side view of MPVPs. Refining a given set of points is performed in an interactive way, by letting the user probe different depths for kD-tree-based refinement and different extents of confidence regions. However, the selection of the best values so that coherent structures in the data are not split up is difficult to achieve. Even though it is clear that a more structure-based refinement can in

principle overcome this problem, this requires a far more involved pre-process to determine coherent (sub-)structures in the multi-dimensional space. Even if such structures can be determined, efficient mechanisms to test whether a data point belongs to such a structure need to be developed first. Concerning the side-by-side view of MPVPs, for ensembles composed of many members this approach is not feasible anymore due to space restrictions. We envision an automatic pre-selection or even grouping of MPVPs so that only few representative members need to be shown at once in a first overview visualization. Finally, we intend to investigate the extension of the workflow to time-varying ensembles, so that the temporal changes of multi-parameter distributions over the ensemble can be assessed and compared in a meaningful way.

ACKNOWLEDGMENTS

This research has been done within the subproject B5 of the Transregional Collaborative Research Center SFB/TRR 165 Waves to Weather funded by the German Research Foundation (DFG).

REFERENCES

- [1] C. Baumgartner, C. Plant, K. Railing, H.-P. Kriegel, and P. Kroger, "Subspace selection for clustering high-dimensional data," in *Proc. 4th IEEE Int. Conf. Data Mining*, 2004, pp. 11–18.
- [2] S. Bergner, M. Sedlmair, T. Moller, S. N. Abdolrousefi, and A. Saad, "ParaGlide: Interactive parameter space partitioning for computer simulations," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 9, pp. 1499–1512, Sep. 2013.
- [3] M. Blumenschein, X. Zhang, D. Pomerence, D. A. Keim, and J. Fuchs, "Evaluating reordering strategies for cluster identification in parallel coordinates," *Comput. Graph. Forum*, vol. 39, pp. 537–549, 2020.

- [4] G.-P. Bonneau *et al.*, "Overview and state-of-the-art of uncertainty visualization," in *Scientific Visualization*. Berlin, Germany: Springer, 2014, pp. 3–27.
- [5] S. Bruckner and T. Möller, "Result-driven exploration of simulation parameter spaces for visual effects design," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1468–1476, Nov./Dec. 2010.
- [6] K. Bunte, S. Haase, M. Biehl, and T. Villmann, "Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences," *Neurocomputing*, vol. 90, pp. 23–45, 2012.
- [7] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2013, pp. 160–172.
- [8] D. B. Carr, R. J. Littlefield, W. Nicholson, and J. Littlefield, "Scatterplot matrix techniques for large N," *J. Amer. Statist. Assoc.*, vol. 82, no. 398, pp. 424–436, 1987.
- [9] J. M. Chambers, *Graphical Methods for Data Analysis*. London, U.K., FL, USA: Chapman and Hall/CRC, 2017.
- [10] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey, *Graphical Methods for Data Analysis*. London, U.K.: Chapman and Hall/CRC, 1983.
- [11] C.-H. Cheng, A. W. Fu, and Y. Zhang, "Entropy-based subspace clustering for mining numerical data," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 1999, pp. 84–93.
- [12] A. Dasgupta, M. Chen, and R. Kosara, "Conceptualizing visual uncertainty in parallel coordinates," *Comput. Graph. Forum*, vol. 31, pp. 1015–1024, 2012.
- [13] I. Demir, C. Dick, and R. Westermann, "Multi-charts for comparative 3D ensemble visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2694–2703, Dec. 2014.
- [14] I. Demir, M. Jarema, and R. Westermann, "Visualizing the central tendency of ensembles of shapes," in *Proc. SIGGRAPH Asia Symp. Vis.*, 2016, Art. no. 3.
- [15] H. Doleisch, M. Mayer, M. Gasser, R. Wanker, and H. Hauser, "Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system," in *Proc. 6th Joint Eurographics - IEEE TCVG Conf. Vis.*, 2004, pp. 91–96.
- [16] H. Doleisch, P. Muigg, and H. Hauser, "Interactive visual analysis of hurricane Isabel with SimVis," *IEEE Vis. Contest*, Citeseer, 2004.
- [17] S. Dutta and H.-W. Shen, "Distribution driven extraction and tracking of features for time-varying data analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 837–846, Jan. 2016.
- [18] B. J. Ferdosi and J. B. Roerdink, "Visualizing high-dimensional structures by dimension ordering and filtering using subspace analysis," *Comput. Graph. Forum*, vol. 30, pp. 1121–1130, 2011.
- [19] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann, "Visual analysis of spatial variability and global correlations in ensembles of Iso-Contours," *Comput. Graph. Forum*, vol. 35, no. 3, pp. 221–230, 2016.
- [20] A. Fofonov, V. Molchanov, and L. Linsen, "Visual analysis of multi-run spatio-temporal simulations using Isocontour similarity for projected views," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 8, pp. 2037–2050, Aug. 2016.
- [21] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Hierarchical parallel coordinates for exploration of large datasets," in *Proc. Conf. Vis.*, 1999, pp. 43–508.
- [22] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Navigating hierarchies with structure-based brushes," in *Proc. IEEE Symp. Inf. Vis.*, 1999, pp. 58–64.
- [23] S. Gama and D. Goncalves, "Studying color blending perception for data visualization," *EuroVis-Short Papers*, N. Elmquist, M. Hlawitschka and J. Kennedy, Eds., The Eurographics Association, 2014, doi: 10.2312/eurovisshort.20141168.
- [24] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Inf. Vis.*, vol. 10, no. 4, pp. 289–309, 2011.
- [25] S. Hazarika, A. Biswas, and H.-W. Shen, "Uncertainty visualization using copula-based analysis in mixed distribution models," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 934–943, Jan. 2018.
- [26] S. Hazarika, S. Dutta, H.-W. Shen, and J.-P. Chen, "CoDDA: A flexible copula-based distribution driven analysis framework for large-scale multivariate data," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 1214–1224, Jan. 2019.
- [27] W. He, J. Wang, H. Guo, H.-W. Shen, and T. Peterka, "CECAV-DNN: Collective ensemble comparison and visualization using deep neural networks," *Vis. Inform.*, vol. 4, pp. 109–121, 2020.
- [28] J. Heinrich and D. Weiskopf, "State of the art of parallel coordinates," in *Proc. Eurogra. State Art Rep.*, 2013, pp. 95–116.
- [29] J. L. Hintze and R. D. Nelson, "Violin plots: A box plot-density trace synergism," *Amer. Statistician*, vol. 52, no. 2, pp. 181–184, 1998.
- [30] T. Höllt *et al.*, "Visual analysis of uncertainties in ocean forecasts for planning and operation of off-shore structures," in *Proc. IEEE Pacific Vis. Symp.*, 2013, pp. 185–192.
- [31] T. Höllt *et al.*, "Ovis: A framework for visual analysis of ocean forecast ensembles," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 8, pp. 1114–1126, Aug. 2014.
- [32] A. Inselberg, "The plane with parallel coordinates," *Vis. Comput.*, vol. 1, no. 2, pp. 69–91, 1985.
- [33] A. Inselberg and B. Dimsdale, "Parallel coordinates," in *Human-Machine Interactive Systems*. Berlin, Germany: Springer, 1991, pp. 199–233.
- [34] M. Jarema, I. Demir, J. Kehr, and R. Westermann, "Comparative visual analysis of vector field ensembles," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2015, pp. 81–88.
- [35] J. Johansson and C. Forsell, "Evaluation of parallel coordinates: Overview, categorization and guidelines for future research," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 579–588, Jan. 2016.
- [36] C. R. Johnson and A. R. Sanderson, "A next step: Visualizing errors and uncertainty," *IEEE Comput. Graphics Appl.*, vol. 23, no. 5, pp. 6–10, Sep./Oct. 2003.
- [37] I. Jolliffe, *Principal Component Analysis*, 2nd ed. Berlin, Germany: Springer, 2010.
- [38] J. Kehr and H. Hauser, "Visualization and visual analysis of multifaceted scientific data: A survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 3, pp. 495–513, Mar. 2013.
- [39] A. Kumpf, J. Stumpfegger, and R. Westermann, "Cluster-based analysis of multi-parameter distributions in cloud simulation ensembles," in *Proc. 24th Int. Symp. Vis. Model. Vis.*, 2019, pp. 73–80.
- [40] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg, "Comparative analysis of multidimensional, quantitative data," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1027–1035, Nov./Dec. 2010.
- [41] L. Linsen, T. Van Long, P. Rosenthal, and S. Rossow, "Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1483–1490, Nov./Dec. 2008.
- [42] S. Liu, J. A. Levine, P.-T. Bremer, and V. Pascucci, "Gaussian mixture model based volume visualization," in *Proc. IEEE Symp. Large Data Anal. Vis.*, 2012, pp. 73–77.
- [43] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 3, pp. 1249–1268, Mar. 2017.
- [44] A. L. Love, A. Pang, and D. L. Kao, "Visualizing spatial multivariate data," *IEEE Comput. Graphics Appl.*, vol. 25, no. 3, pp. 69–79, May/Jun. 2005.
- [45] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
- [46] J. D. MacDonald and K. S. Booth, "Heuristics for ray tracing using space subdivision," *Vis. Comput.*, vol. 6, no. 3, pp. 153–166, 1990.
- [47] V. Molchanov and L. Linsen, "Overcoming the curse of dimensionality when clustering multivariate volume data," in *Proc. 13th Int. Joint Conf. Comput. Vis. Imag. Comput. Graph. Theory Appl.*, 2018, pp. 29–39.
- [48] A. Morrison, G. Ross, and M. Chalmers, "A hybrid layout algorithm for sub-quadratic multidimensional scaling," in *Proc. IEEE Symp. Inf. Vis.*, 2002, pp. 152–158.
- [49] J. E. Nam and K. Mueller, "TripAdvisor {ND}: A tourism-inspired high-dimensional space exploration framework with overview and detail," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 2, pp. 291–305, Feb. 2013.
- [50] H. Obermaier and K. I. Joy, "Future challenges for ensemble visualization," *IEEE Comput. Graphics Appl.*, vol. 34, no. 3, pp. 8–11, May/Jun. 2014.
- [51] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf, "An edge-bundling layout for interactive parallel coordinates," in *Proc. IEEE Pacific Vis. Symp.*, 2014, pp. 57–64.
- [52] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha, "Approaches to uncertainty visualization," *Vis. Comput.*, vol. 13, no. 8, pp. 370–390, 1997.

- [53] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson, "Visualizing summary statistics and uncertainty," *Comput. Graph. Forum*, vol. 29, no. 3, pp. 823–832, 2010.
- [54] R. G. Raidou, M. Eisemann, M. Breeuwer, E. Eisemann, and A. Vilanova, "Orientation-enhanced parallel coordinate plots," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 589–598, Jan. 2016.
- [55] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *Proc. 5th Int. Conf. Coordinated Multiple Views Explor. Vis.*, 2007, pp. 61–71.
- [56] R. C. Roberts, R. S. Laramée, G. A. Smith, P. Brookes, and T. D'Cruze, "Smart brushing for parallel coordinates," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 3, pp. 1575–1590, Mar. 2019.
- [57] H. Saikia and T. Weinkauf, "Global feature tracking and similarity estimation in time-dependent scalar fields," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 1–11, 2017.
- [58] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead, "Noodles: A tool for visualization of numerical weather model ensemble uncertainty," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1421–1430, Nov./Dec. 2010.
- [59] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller, "Visual parameter space analysis: A conceptual framework," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2161–2170, Dec. 2014.
- [60] R. Swinbank *et al.*, "The TIGGE project and its achievements," *Bull. Amer. Meteorol. Soc.*, vol. 97, no. 1, pp. 49–67, Mar. 2015.
- [61] D. Thompson *et al.*, "Analysis of large-scale scalar data using hixels," in *Proc. IEEE Symp. Large Data Anal. Vis.*, 2011, pp. 23–30.
- [62] T. Torsney-Weir *et al.*, "Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 1892–1901, Dec. 2011.
- [63] T. Van Long and L. Linsen, "MultiClusterTree: Interactive visual exploration of hierarchical clusters in multidimensional multivariate data," *Comput. Graph. Forum*, vol. 28, pp. 823–830, 2009.
- [64] J. Wang, S. Hazarika, C. Li, and H.-W. Shen, "Visualization and visual analysis of ensemble data: A survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 9, pp. 2853–2872, Sep. 2019.
- [65] K.-C. Wang, K. Lu, T.-H. Wei, N. Shareef, and H.-W. Shen, "Statistical visualization and analysis of large data using a value-based spatial distribution," in *Proc. IEEE Pacific Vis. Symp.*, 2017, pp. 161–170.
- [66] M. O. Ward, "XmdvTool: Integrating multiple methods for visualizing multivariate data," in *Proc. Conf. Vis.*, 1994, pp. 326–333.
- [67] C. Wellmann *et al.*, "Using emulators to understand the sensitivity of deep convective clouds and hail to environmental conditions," *J. Adv. Model. Earth Syst.*, vol. 10, no. 12, pp. 3103–3122, 2018.
- [68] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, "Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2713–2722, Dec. 2013.
- [69] M. Williams and T. Munzner, "Steerable, progressive multidimensional scaling," in *Proc. IEEE Symp. Inf. Vis.*, 2004, pp. 57–64.



Alexander Kumpf received the BSc degree in mathematics and the MSc degree in mathematical finance from the Technical University of Munich (TUM), Munich, Germany, in 2012 and 2015, and the PhD degree in computer science from TUM, Munich, Germany, in 2020. His major interests in research are visualization of ensemble simulation data using statistical analysis, machine learning, and neural-network-based data improvement.



Josef Stumpfegger received the BSc degree in computer science from the Technical University of Munich, Munich, Germany, in 2019, where he is currently working toward the undergraduate degree. His major interests in research are scientific data visualization for large-scale data, real-time rendering, and high performance GPU computing.



Patrick Fabian Härtl received the BSc and MSc degrees in computer science/games engineering from the Technical University of Munich, Munich, Germany, in 2017 and 2020, respectively. His research interests include real-time rendering, ray tracing, scientific visualization, and extended reality (XR).



Rüdiger Westermann received the degree in computer science from the Technical University Darmstadt, Darmstadt, Germany, and the PhD degree in computer science from the University of Dortmund, Dortmund, Germany. In 2002, he was appointed the chair of Computer Graphics and Visualization, TUM. His research interests comprise scalable data visualization and simulation algorithms, GPU computing, real-time rendering of large data, and uncertainty visualization.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.