Vision, Modeling, and Visualization (2019)
H.-J. Schulz, M. Teschner, and M. Wimmer (Eds.)

Cluster-based Analysis of Multi-Parameter Distributions in Cloud Simulation Ensembles

Alexander Kumpf, Josef Stumpfegger, Rüdiger Westermann

Technical University of Munich (TUM), Germany

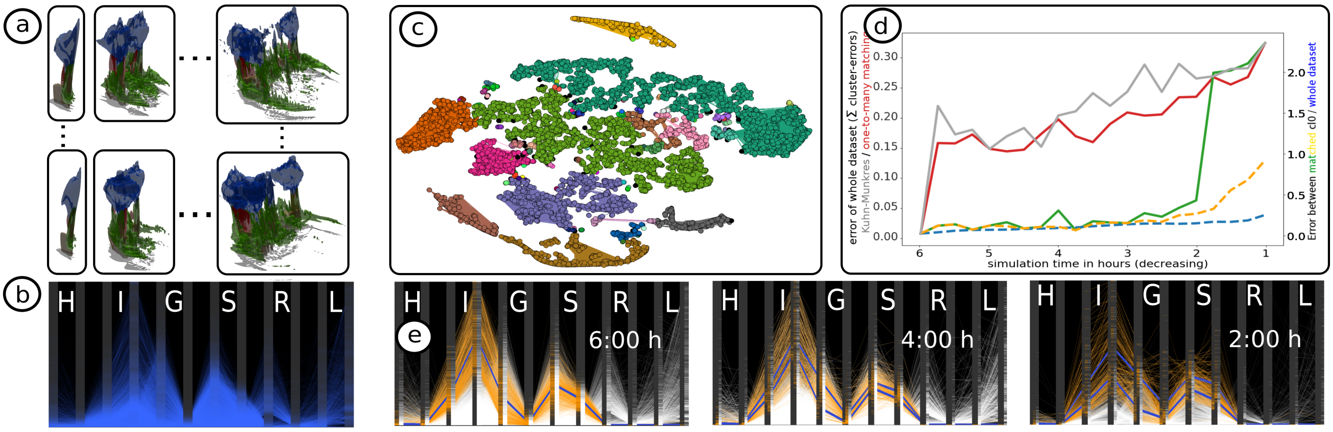

Figure 1: *Visual analysis of a time-varying multi-parameter cloud ensemble. (a) Iso-surfaces of liquid water content (L, green), ice (I, blue), and graupel (G, red). (b) Parallel coordinates plot rendered on black background of per-voxel parameter values including attribute density histograms. (c) Voxel clusters determined via ensemble clustering using multiple k-Means clustered t-SNE projections. Voxels in the same cluster but separated in the t-SNE embedding are connected via lines. (d) Distribution-based matching of clusters over lead time, starting from 6 hours forecast. Colored curves show the matching error for different matching strategies and error metrics. (e) Selected cluster at 6 hours lead time (orange) and clusters matched to it at 4 and 2 hours lead time are highlighted in the parallel coordinates plot.*

Abstract

The proposed approach enables a comparative visual exploration of multi-parameter distributions in time-varying 3D ensemble simulations. To investigate whether dominant trends in such distributions occur, we consider the simulation elements in each dataset—per ensemble member and time step—as elements in the multi-dimensional parameter space, and use t-SNE to project these elements into 2D space. To find groups of elements with similar parameter values in each time step, the resulting projections are clustered via k-Means. Since elements with similar data values can be disconnected in one single projection, we compute an ensemble of projections using multiple t-SNE runs and use evidence accumulation to determine sets of elements that are stably clustered together. We build upon per-cluster multi-parameter distribution functions to quantify cluster similarity, and merge clusters in different ensemble members. By applying the proposed approach to a time-varying ensemble, the temporal development of clusters, and in particular their stability over time can be analyzed. We apply this approach to analyze a time-varying ensemble of 3D cloud simulations. The visualizations via t-SNE, parallel coordinate plots and scatter plot matrices show dependencies between the simulation results and the simulation parameters used to generate the ensemble, and they provide insight into the temporal ensemble variability regarding the major trends in the multi-parameter distributions.

1. Introduction

Ensemble weather forecasting is well established in meteorology to estimate the uncertainty that is present in numerical weather predictions. Ensemble methods perform multiple simulations using perturbed initial conditions or different forecast models, to predict possible future states of the atmosphere. Analysis of the temporal evolution and variability of an ensemble forecast is then used to estimate the likelihood of certain weather events.

Ensemble methods are also used to analyze the effect of simulation parameters on the simulated weather events, by systematically

DOI: 10.2312/vmv.20191320

perturbing these parameters and running the simulation again using the perturbed configurations. The analysis, then, requires more than detecting similarities and differences between pairs of simulation results. Beyond that, coherent predictions, localized or over the entire domain, across many results need to be determined and put into relation to the input parameter values that have caused these situations. For an improved assessment of these predictions, they need to be quantified and visualized in context to each other.

We propose a visual analysis approach to help addressing this task, and use this approach to analyze an ensemble of time-varying 3D cloud simulations [WBJ*18]. Each simulation is carried out on a regular voxel grid, and for each voxel a 12D parameter vector containing quantities like water, ice, graupel, and hail content is simulated. As we seek to compare clouds of different size, shape and position, location-based approaches, i.e., computing statistical measures over the values at a single voxel, are not useful. For instance, if a cloud does not change with respect to its physical composition but simply moves in space, similarity measures invariant under such transformations need to be used.

Parallel coordinate (PC) plots can, in principle, be used for this purpose, by drawing one line strip for each voxel. However, it is difficult to reveal lower-dimensional manifold structures via PC plots, and the visualization becomes quickly cluttered when many elements are drawn simultaneously. Clustering, on the other hand, can determine sub-groups of elements with similar parameter values, providing a condensed data representation that facilitates a distribution-based analysis. Clustering in high-dimensional (HD) parameter space, however, becomes difficult due to the inherent sparsity of the data space and the difficulty to select an appropriate clustering algorithm and its parametrization.

Dimensionality reduction techniques can be used to address these problems. For instance, t-Distributed Stochastic Neighbor Embedding (t-SNE) [MH08] tries to preserve locality by placing similar elements close to each other in a low-dimensional subspace. Since lower-dimensional manifold structures in the original data are preserved, especially density-based clustering algorithms like DBSCAN [EKS*96], which focus on “reachability” rather than distance, show very good results if the right parametrization is used. However, since the parametrization needs to be adapted for every projection, in our current scenario the application of DBSCAN is not feasible.

1.1. Contribution and method overview

The proposed approach detects stable clusters of data points in a HD parameter space. It uses this information to enable a cluster-based analysis of the variability of ensembles of multi-parameter simulations, and to reveal dependencies of the simulation results on the initial simulation parameters. An overview of this approach is given in Fig. 2. By variations of a set of input parameters τ_i, an ensemble of multi-parameter simulations is computed. Simulation elements are interpreted as data points in the multi-dimensional parameter space, and they can be visualized using standard visualization techniques like volume rendering and PC plots.

Then, dimensionality reduction via t-SNE projects the data points into 2D. In this way, many of the local neighborhoods in the data are preserved and sub-manifolds in HD space become connected structures in 2D space. To avoid the shortcomings of DBSCAN in the current scenario, the projected points are clustered using k-Means, and the resulting clusters are put into relation using their variability over the ensemble.

Dimensionality reduction techniques like t-SNE, however, sometimes need to split a connected subgroup to compute the 2D embedding. Where these splits occur depends on the specific parametrization of the used technique. For instance, t-SNE is often used with random initial locations of projected objects as its seed configuration which are considered by gradient descent optimization. Therefore, when t-SNE is run with different input parameterizations, connected subgroups can be split in many different ways.

On the other hand, similar points should be placed close to each other most of the time over all projections, regardless of the specific initial parametrization. Thus, we compute many projections using different parameterizations and merge the clusterings which are obtained via k-Means into one final clustering. To visualize stable subgroups, the projection representing best the final clustering is picked, and cluster membership information per data point is encoded via colors. Additionally, some of the points are connected via lines to indicate where clusters were cut in the selected projection but can be assumed connected over all projections. To further analyze the distribution of parameter values in a cluster, they are displayed via PC plots, augmented by per-parameter distributions and overlayed representatives for selected clusters. Per-cluster distributions are then represented via cumulative distribution functions (CDF), and the differences in their integrals are used as similarity measure. This enables to match different clusters and find similarities across time steps and ensemble members. The similarity between ensemble members is put in context with the initial simulation parameters via scatter plot matrices.

In particular, the following contributions are made:

- A method to determine stable clusters in multi-parameter data sets, using t-SNE and k-Means-based ensemble clustering.
- A distribution-based similarity metric for clusters of multi-parameter data points.
- The application of cluster-based analysis of multi-parameter distributions to a time-varying multi-parameter 3D cloud ensemble, hinting on the effect of simulation parameters on weather forecast variability.

On a technical side, we provide a highly efficient GPU implementation of PC plots with embedded line and density histograms capable of plotting millions of multi-parameter data elements per second, including instant color variations to highlight selected clusters. In the 2D t-SNE view, multiple interaction possibilities are available to select and display single clusters, similar data elements, etc., over different projections, time steps, and ensemble members.

2. Related work

In our scenario, each ensemble member is comprised of a set of simulation elements with multiple parameter values. These HD data points are projected into 2D using t-SNE [MH08]. Some recent surveys [KH13; LMW*16] give thorough overviews of visualization techniques for multi-parameter data. In combination with

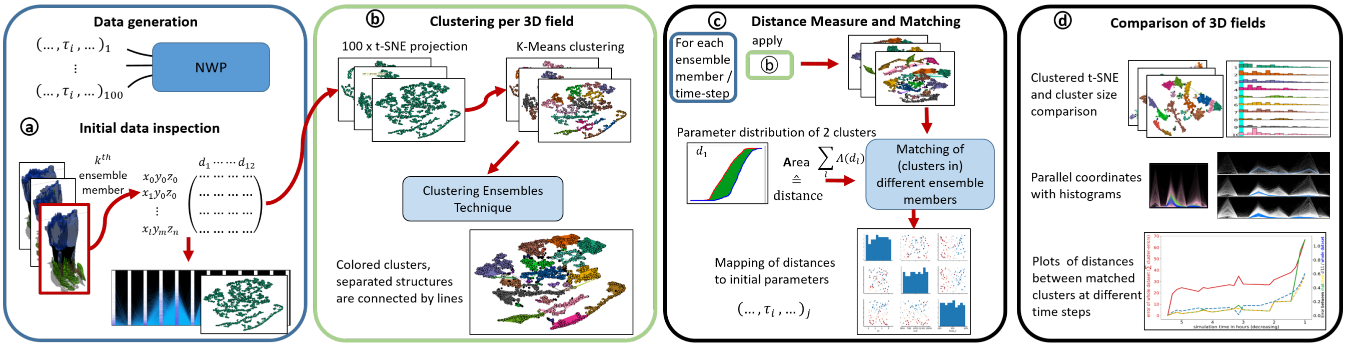
Data generation
$(\dots,\tau_i,\dots)_1$
⋮
$(\dots,\tau_i,\dots)_{100}$
NWP
ⓐ Initial data inspection
k^{th} ensemble member
$x_0y_0z_0$
$x_1y_0z_0$
⋮
$x_ly_mz_n$
$d_1 \cdots\cdots d_{12}$
ⓑ Clustering per 3D field
100 x t-SNE projection
K-Means clustering
Clustering Ensembles Technique
Colored clusters, separated structures are connected by lines
ⓒ Distance Measure and Matching
For each ensemble member / time-step
apply ⓑ
Parameter distribution of 2 clusters
d_1
Area ≙ distance
$\sum_i A(d_i)$
Matching of (clusters in) different ensemble members
Mapping of distances to initial parameters
$(\dots,\tau_i,\dots)_j$
ⓓ Comparison of 3D fields
Clustered t-SNE and cluster size comparison
Parallel coordinates with histograms
Plots of distances between matched clusters at different time steps
simulation time in hours (decreasing)

Figure 2: *Proposed workflow for analyzing an ensemble of 3D multi-parameter simulations, each simulation parameterized by input parameters τ_i. **(a)** Upon using standard visualization techniques like volume rendering and PC plots, **(b)** k-Means clusterings on multiple t-SNE projections are combined to obtain a stable clustering of data points in a single 3D data set. The best embedding is determined for visualization. **(c)** For all ensemble members and time steps, clusters are matched based on their CDFs. A scatter plot matrix indicates relations between input parameters and multi-parameter distributions. **(d)** Visualization of the (temporal) variability of simulations using clustered t-SNE, cluster bar charts, PC plots, histograms, and line charts.*

dimensionality reduction, clustering is often used to identify groups of points lying close together in the low-dimensional space or forming coherent structures in this space. Wenskovitch et al. [WCR*18] discuss the combination of dimensionality reduction and clustering techniques and provide recommendations for their concurrent use.

Since our technique analyzes an ensemble of 2D point sets (after multi-parameter simulation elements have been projected into 2D), it is related to ensemble visualization techniques. Most works in ensemble visualization address ensembles of physical fields, or features derived from such fields, with the focus on the extraction and visual encoding of their variability. To the best of our knowledge, visual analysis techniques for ensembles of 2D points are not existing, yet a number of techniques have addressed aspects related to ensembles that are also relevant in our work. Parametric statistical distributions and distribution shape descriptors for scalar-valued ensembles were presented by Love et al. [LPK05]. Different variants of confidence regions were introduced to represent the major geometric trends in ensembles of iso-contours and streamlines [WMK13; MWK14; FBW16; FKRW16]. Demir et al. [DJW16] proposed a closest-point representation to convey the central tendency of an ensemble of multi-dimensional shapes. In a number of works, scalar- and vector-valued ensemble fields were modeled via mixtures of probability density functions to compactly classify complex distributions and their evolution over time [LLBP12; JDKW15; DS15; WLW*17]. Demir et al. [DDW14] visualize distributions of linearized 3D data points with bar-charts. Hummel et al. [HOGJ13] analyze the spread of particle trajectories in an ensemble of vector fields to reveal the transport variability. Poethkow and Hege [PH13] and Athawale et al. [ASE15] use location-wise estimators of non-parametric distributions from ensemble members to estimate the spread of surface and vector field features. Recently, Hazarika et al. [HBS17; HDSC19] presented a copula-based framework for large multivariate datasets, where they partition the domain and compute statistical quantities over those parts.

Alternatively, clustering has been used to group ensemble members regarding similar data characteristic [BM10; OLK*14; FBW16]. While these techniques compare ensemble members to each other, our approach aims at finding groups of elements in each member which remain "close" to each other in all members, and then match different clusterings to each other. Strehl and Ghosh [SG02] apply different clustering techniques to one single ensemble, and combine the results into a single clustering. Different clustering ensemble techniques, i.e., techniques that combine multiple clusterings of one data set into a single clustering, are discussed by Vega-Pons et al. [VR11]. From multiple k-Means clusterings, Fred and Jain [FJ05] generate a co-association matrix, containing the fraction of times two points were placed into the same cluster. Applying clustering on this matrix leads to the final result. Kumpf et al. [KTB*18] use multiple k-Means clusterings on ensemble data, where they vary the clustering domain to generate a clustering ensemble. Ferstl et al. [FKRW17] cluster different time steps of the same ensemble in a hierarchical way to convey the change of clusters over time. For the clustering of genomic data, Lex et al. [LSP*10] introduce extended PC plots to compare different clusterings and analyze the quality of cluster assignments.

Related to our approach are also techniques which aim to find projections that best represent the structures in HD data, by using quality measures for projections [FT74; HA85]. Even though the goal of these techniques is different to ours, as we do not attempt to find the best projection for a given dataset, proposed measures indicate the (dis-)similarity between projections and might be used for robustness analysis as well. Examples include vector distance measures for HD feature descriptors [BvLBS11] and feature vectors derived from point-wise distance matrices [JHB*17], as well as measures using matrix norms to quantify the dissimilarity of multivariate projections invariant to affine transformations [LT16].

3. Data

We apply our cluster-based approach to analyze the multi-parameter distributions in a numerical simulation of a growing thunderstorm cloud [WBJ*18]. The data set comprises an ensemble of 100 simulation runs of a single convective cloud in the 3D atmosphere, simulated over a time span of 6 hours in time steps of 15 minutes. 4 members have been excluded due to corrupted values resulting in a total of 96 members. At each 3D position, 12 parameters—such as water content, ice-water content, number of water particles, number of ice particles—are given. The numerical simulation depends on 6 input parameters such as wind-sheer, which influence the outcome of the simulation.

Due to computational reasons, only every 4$^{\text{th}}$ value in the two spatial coordinates x and y, and every 3$^{\text{rd}}$ value in the vertical is used, resulting in roughly 10000 data points per time step. Values of each attribute are normalized to the range of $[0,1]$ excluding the ones with norm smaller 0.1 in order to shift the focus of the analysis away from almost empty voxels. No restrictive assumptions about the structure of clusters are made. Further, since quantities within clouds transition smoothly between states, as water slowly starts to freeze with decreasing temperature, elongated structures are expected at the least.

4. Dimensionality reduction and k-Means clustering

For dimensionality reduction, the method t-SNE (Fig. 4a) is used. Note that while variants such as Hierarchical Stochastic Neighbor Embedding [PHL*16] can be used as well, deterministic dimensionality reduction techniques like principal component analysis (PCA) are not suitable in the current scenario. Points can be misplaced due to variation in others than the principal components subspace used for projecting. Re-running PCA would not change that. Multi-dimensional scaling [KW78], on the other hand, seeks at preserving distances over the whole domain, thus making it difficult to maintain local structures in the generated 2D embeddings.

In a single projection, the distances between data points can be significantly distorted depending on the parametrization of the used projection technique. The reason is that dimensionality reduction techniques need to cut manifold structures in the HD space to embed them into 2D. For instance, when projecting a sphere there is no 2D embedding that can avoid placing non-neighboring points close together or flattening the sphere so that neighboring points become distant to each other. This problem can be addressed by running t-SNE many times with different parameters or random initialization, so that cuts are introduced at different locations and the neighborhood relations are maintained in most projections. Each projection can be clustered individually, and the clustering results can be further analyzed to extract sets of data points that are coherently assigned to the same cluster. In addition, however, the individual clusterings need to create consistent results for different ensemble members and time steps, to allow for a later comparison of these results. It is clear that this cannot be achieved by tweaking the parameters of each individual clustering. Due to this requirement, density-based clustering approaches (e.g., DBSCAN [EKS*96]) are not suitable in the current application. The clustering results of these algorithms are rather sensitive to variations in the distances between projected data points, which, as described before, can happen to a certain extent in different t-SNE projections. It is worth noting that the same problem occurs when clustering is applied to the original HD data point, as shown in Fig.3a.

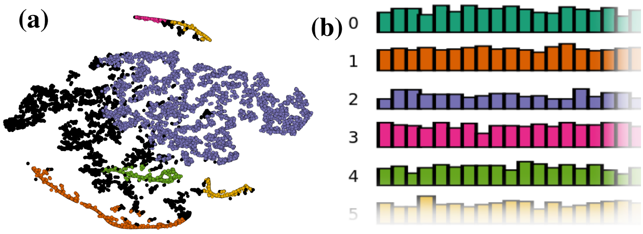

Figure 3: ***(a)*** *High-dimensional DBSCAN clustering with parameters* $\varepsilon = 0.04$ *and* $N = 30$ *color coded on t-SNE projected points. The blue cluster and noise in black dominate the clustering.* ***(b)*** *Matched clusters of k-Means clustered t-SNE projections. Each column represents one clustering, the height of the bar encodes the number of points contained in a cluster.*

The clustering algorithm k-Means, on the other hand, always generates a predefined number of clusters. Though, the clusters are convex and can hence put two independent elongated structures into the same cluster or cut them at an arbitrary location. However, in different projections these cuts are introduced at different locations; furthermore, if these structures are not adjacent in the original data, a different t-SNE projection is likely to place them far apart from each other in the computed 2D embedding. Therefore, only points that are neighbored in the HD space should be in the same cluster in most of the projections, thus overcoming the convexity requirement of k-Means clusters.

Due to the aforementioned issues, we use t-SNE, with default perplexity of 30, and k-Means, with $k = 16$ clusters, in our analysis. The perplexity parameter controls the size of the local neighborhood that should be preserved. In all of our experiments, the resulting projections looked reasonable, showing frequent yet spurious variations that support our envisioned consistency analysis. The number of clusters for k-Means has to be set in relation to the number of projections used. The higher the number of different t-SNE projections, the more clusters can be used to obtain more detailed results. The same parameters are used for all ensemble members and time steps in order to preserve comparability.

4.1. Combination of clusterings

The ensemble of clusterings that is generated by clustering multiple t-SNE projections separately is aggregated to obtain a final clustering. Points that are stably clustered together are extracted by using the so-called co-association matrix C [FJ05]. Each entry, C_{ij} counts how often point p_i and p_j are in the same cluster, finally normalized by dividing through the number of clusterings. For every point, the clique of points with high mutual similarity is searched in C. The similarity threshold is set to $\alpha = 0.9$, meaning that every pair of points in the same clique is clustered together in at least 90% of the single k-Means clusterings. For clique construction, we use algorithm 1 as proposed in Kumpf et al. [KRRW19]. In a final step, illustrated in Fig. 5, points are merged based on their cliques in a greedy-like manner using region growing. Starting with the point with the largest clique, recursively, all points therein and in their cliques are merged. Once no more points can be added, a cluster is formed and recursive merging is continued with the remaining points, starting again with the one with the largest clique.

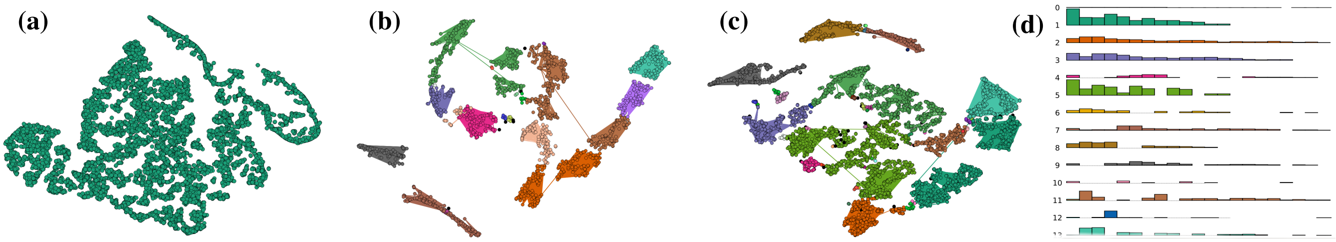

Figure 4: *(a) 2D t-SNE projection of voxels. Connected structures motivate the use of clustering. (b-c) Final clustering resulting from the combination of* 100 *k-Means clustered t-SNE projections. Lines connect distant points in clusters, points of the same cluster have the same color, with noise in black. The projections with shortest line sums were chosen for the 2D embedding. Time steps **(b)** 2:00h **(c)** 4:00h are shown. **(d)** Each column represents a clustering of one time step starting with the latest, where height encodes the cluster size.*

```
input: $i_0, C, \alpha$    output: $L$
$i = i_0$;   $I = \{0, \dots, n\}$;   $L = \{\}$;
while ($|L| < |I|$) do
    $L = L \cup \{i\}$;          $I = I \setminus \{i\}$;
    $I = \{j \in I | C_{ij} > \alpha\}$;   $i = \arg\max_{j \in I} C_{ij}$;
end
```

Algorithm 1: Generation of point clique for p_{i_0} (similar to [KRRW19]) using the co-association matrix C using a pairwise similarity threshold α. Resulting cliques contain points with mutual similarity greater than α.

The merging algorithm depends on the order in which points are traversed. However, using random initial points has lead to similar results in all our experiments. To better understand the merging process, k-Means clusters can be matched as in [KRRW19] over all projections using the Kuhn-Munkres algorithm [Kuh55] (Fig. 3b). Single clusters can be selected and their position is displayed over different t-SNE projections. Fixing the projection and showing clusters from other projections is available as well, revealing neighboring points in other projections which were placed apart in the current one. Furthermore, one can search for points which were always assigned to a selected cluster or highlight points which were almost always together in the same cluster, which greatly helps understanding the effect of the merging threshold α. These interactions help understanding the quality and variance of the clustering ensemble and can be performed before comparing datasets. Later, it can be used to see the evolution of single clusters or understand why certain structures fall into different clusters.

The final obtained clusters represent points which lie in the same structure in the t-SNE plot and are therefore expected to form structures in HD parameter space. An example is given in Fig. 1c. Note that the number of clusters can now exceed 16. Points are colored according to their cluster ID, using black for noise. Additionally, lines between points, and in the color of these points, are drawn if the points are far apart. Short lines can be filtered out interactively.

Lines connecting adjacent points but located far away from each other in the current projection are generated as a byproduct during the merging step. Whenever points are merged, lines from the parent to all children are saved and used later in the final clustering. This facilitates the identification of clusters which were torn apart in the currently selected projection and attenuates the problem of finding a sufficiently large number of distinguishable cluster colors, i.e., clusters that are far apart from each other and not connected are different regardless their assigned color.

To visualize the cluster information, one projection has to be selected as a representative 2D embedding of the data points. We use a projection instead of other cluster visualizations, since these projections preserve the spatial relationships between points and clusters. Following the intuition that points in clusters should be located close to each other, we select the projection with the minimal sum of filtered line-lengths between the data-points. The final clustering result is then investigated further using PC plots, i.e., to compare different time steps and ensemble members.

4.2. Matching and comparing clusterings

A relation between final clusters of different ensemble members is established by comparing the distributions of their parameter values. For each parameter, a CDF is constructed [HDSC19]. To compare two clusters, for each parameter the area between their CDFs is computed (see Fig. 6) and summed up. Since this similarity measure depends only on the distribution of the parameter values, it can be used to compare clusters with vastly different size. Since this could become as extreme as matching two points to the biggest clusters, we penalize differences of a factor 10 and higher by adding a linear factor of

$$\text{penalty}_{cl_i, cl_j} = \max\left(0, \left(\frac{\max(|cl_i|, |cl_j|)}{\min(|cl_i|, |cl_j|)} - 10\right) \cdot 0.01\right),$$

to the cluster distance. Here, $|cl_i|$ denotes the size of cluster i. This similarity measure can be used to determine the similarity between two data sets, i.e., two ensemble members or different time steps of the same member, and to compare two clusters.

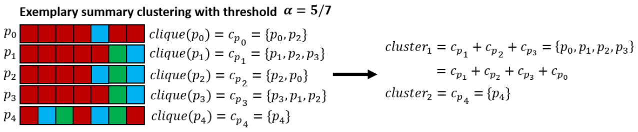
Exemplary summary clustering with threshold $\alpha = 5/7$
p_0
p_1
p_2
p_3
p_4
$clique(p_0) = c_{p_0} = \{p_0, p_2\}$
$clique(p_1) = c_{p_1} = \{p_1, p_2, p_3\}$
$clique(p_2) = c_{p_2} = \{p_2, p_0\}$
$clique(p_3) = c_{p_3} = \{p_3, p_1, p_2\}$
$clique(p_4) = c_{p_4} = \{p_4\}$
$cluster_1 = c_{p_1} + c_{p_2} + c_{p_3} = \{p_0, p_1, p_2, p_3\}$
$= c_{p_1} + c_{p_2} + c_{p_3} + c_{p_0}$
$cluster_2 = c_{p_4} = \{p_4\}$

Figure 5: *Clustering to combine the clustering ensemble. Color denotes the cluster ID for each point p_i for 7 clusterings. Note that clique(p_2) does not contain p_1 as $sim(p_0, p_1) < 5/7$. For this example only, the clique threshold was set to $\alpha = 5/7$.*

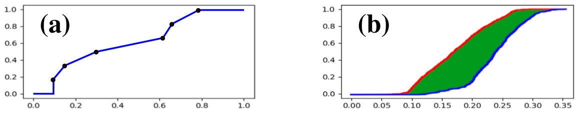

Figure 6: ***(a)*** *Construction of a Cumulative Distribution Function (CDF) based on* 6 *sample points.* ***(b)*** *Comparison of the parameter distribution of 2 clusters (red and blue) by constructing a CDF. The area between the lines defines the distance between the clusters.*

Clusters are then matched using either the Kuhn-Munkres-algorithm [Kuh55], resulting in a one-to-one matching between $\min(|cl_i|,|cl_j|)$ clusters, or alternatively, in a one-to-many matching, where each cluster of one clustering is matched to the best fitting cluster of the other clustering. For single clusters, the results match most of the time as can be seen by the green and yellow line in Fig. 1d. However, mismatches can lead to significantly higher differences, which is why we favor the one-to-many matching.

Summing over the $ln(|cl_i|)$-weighted differences between clusters and normalizing them by the summed weights serves as a second distance measure for data sets (Fig. 1d,red). When multiple *clusterings* are matched to one reference (Sec. 4.4), the sizes of the clusters in the reference clustering are used. Cluster sizes are not important to us. However, since the number of clusters should not affect the measure significantly, less weight is given to smaller clusters to prevent them from dominating the measure. Alternatively, only the x largest matched clusters can be used to compute the distance measure. In our experiments, both strategies are used, where in the latter clusters with less than 25 points are not matched.

4.3. Parallel Coordinates

Based on the matching errors, the sizes of clusters over time (see Fig. 4d), and by using PC plots, data sets can be compared to each other and similarities in parameter distributions can be investigated.

PC plots offer a direct visualization of HD data points. Our implementation uses the Vulkan graphics API, to enable the efficient visualization of huge numbers of multi-parameter data points. On our target architecture, an NVIDIA GTX 1070, up to 5 million 12D data points can be drawn per second. Basic functionality like blending and the reordering of axes can be used to get a first impression of the data. Histograms per displayed cluster on the coordinate axes ease the comparison of value distributions. Optionally, lines can be smoothed to better show densities. Mean and median lines can be drawn instead of whole clusters to avoid visual clutter.

4.4. Selection of reference

The presented analysis requires a reference dataset as starting point, to which others are compared to. Commonly, the simulation generated with best guessed initial parameters is used for that purpose, which is unfortunately not know for this dataset. Instead, the simulation generated with the median of all initial parameter configurations is investigated first. All initial parameter configurations are displayed in a scatter-plot-matrix (Fig. 8) were their distributions can be seen. The matching distances to the selected reference are displayed in color indicating which parameter changes lead to larger distances between the data sets. To gain an understanding for the data set, different members can be selected as reference to further investigate dependencies with initial parameters.

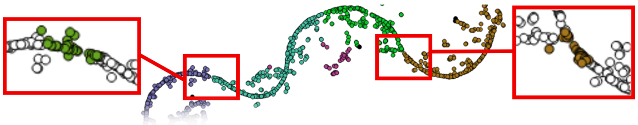

Figure 7: *Reducing the merging threshold* α *to* 0.8 *creates connections between the clusters which leads to a merge.*

5. Use case

In the following, we describe the application of our approach to analyze the multi-parameter cloud ensemble described in Sec. 3. At the beginning, multiple time steps of a single ensemble member are visualized using iso-surfaces in single parameter fields (Fig. 1a). Despite the inherent occlusion effects, the overall structure of the clouds can already been observed: While wet quantities like liquid water (green, L) or rain (yellow, R) dominate in the lower altitudes, frozen quantities like ice (blue, I), graupel (red, G) and hail (brown, H) dominate in the upper atmosphere levels.

The shapes of the clouds change significantly over time, and they move over the domain, so that location-wise computation and comparison of data statistics is no option. Instead, we abstract from the 3D shape and perform the analysis using the distribution of parameter values as described. Firstly, PC plots are generated to obtain an initial estimate of the parameter distributions (Fig. 1b). By looping through the plots of all time steps of a selected ensemble member, the distribution variability over time is conveyed. The distributions seem to stay similar over all time steps, with the exception of strong hail (H), which is present only in later time steps. This is expected since ice-particles need some time to grow within the cloud. However, it is difficult to see whether the cloud forecasts are comprised of individual structures. To analyze this, the data is projected using t-SNE (Fig. 4a). The projections of simulation elements into 2D reveal many band structures and clusters of elements, yet it is impossible to conclude on which structures belong together and which are separated. After generating a stable clustering (explained in Sec. 4 and 4.1), clearly separated clusters appear (color coded in Fig. 1c). Connecting lines highlight where these structures where not cut in other projections, e.g., the rose cluster. Furthermore, small blue clusters of almost the same color can be differentiated, none of them connected via lines.

To investigate which clusters might merge due to a different merging threshold α, points can be picked interactively on the boundary between clusters and the effect of varying α can be seen (Fig. 7). When points from both sides pop-up, the clusters would merge. In this way, the cluster ensemble step and the degree of dissimilarity between clusters can be better understood. Further interaction mechanisms, e.g., selection and tracing of clusters, are provided as additional options to the user.

By using the proposed approach, points of specific clusters can be directly emphasized in the PC plot (Fig. 1e). For every quantity, there are two axes showing the weight and number of particles of that quantity in the corresponding simulation element. Elements in the orange cluster contain mostly ice (I), snow (S) and graupel (G).

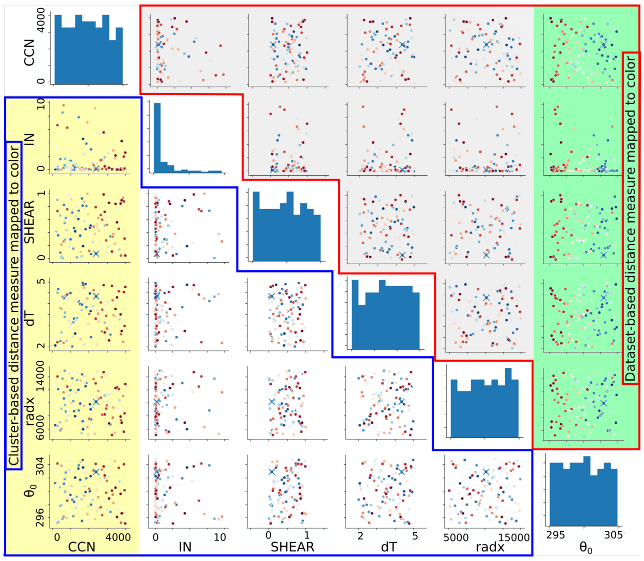

Figure 8: *Matching distances to median simulation based on whole data set and on clusters, color coded on initial parameters, with low-to-high in blue-to-red. The data set- and cluster-based distance measures reveal correlations with* θ_0 *or CCN respectively.*

Next, we cluster all time steps of the selected ensemble member and match clusters to the last by using the proposed many-to-one matching (see Sec.4.2). The matching errors (Fig. 1d) indicate that until time step 2:00h the distributions are very similar and the matching works well. For earlier time steps, the errors grow. Closer inspection of the worst matched clusters reveal that precipitation parameters are differently distributed in earlier time steps. When looping through the colored t-SNE plots—with matched clusters, similar structures can be observed over different time steps (Fig. 4). Caused by the many-to-one matching, some clusters become empty when no matching partner could be found (Fig. 4d). By focussing on the orange cluster to which two clusters were matched at time step 2:00h (Fig. 4b), its time evolution can be displayed using PC plots (Fig. 1e). The median or mean lines (blue lines) as well as a histogram bar per cluster and axis can be selected as well. It can be observed that the distribution of the cluster stays mainly the same, while the number of simulation elements decreases over time.

A similar analysis can be performed over all ensemble members. The proposed metrics can be used to find similar and dissimilar members. First results can be seen in Fig. 8, where matching distances are color coded on the initial parameters. A correlation between temperature (θ_0) and the distance between the whole data sets is directly visible and highlighted in green. When using the cluster based measure, correlation with cloud condensation nuclei (CCN), highlighted in yellow, becomes visible. This indicates that θ_0 changes the overall distributions while CCN changes the structures in the parameter space. Since all initial parameters were perturbed simultaneously, multi-dimensional dependencies have to be considered as well, which is left for future work.

Computation time: Performance is measured on an Intel® Xeon CPU 6 cores @3.5GHz. Preparing each data set takes around 35s, 70s for each t-SNE projection, and 0.5s for k-Means, all performed on one core. Combining the cluster ensemble takes 240s using all cores and cluster matching around 15s per dataset on one core. These pre-computations can be parallelized over the datasets.

6. Discussion and conclusion

Many steps of the proposed approach are dependent on parameters, albeit most of them are rather uncritical when kept constant over the whole analysis. Together, the number of k-Means clusters and projections define the granularity of the approach. More projections allow more k-Means clusters, the ratio of 100 projections to 16 clusters resolved the structures quite well in the presented data set. Multiple k-Means clusterings per projection could further reduce the number of projections needed.

The most critical parameter is the matching threshold α. Its impact can be seen in Fig. 7, where reducing α from 0.9 to 0.8 would merge the structure. A smaller threshold leads to bigger clusters. We advice to chose and fix this parameter once in the beginning after fixing all other parameters. That way, the merging stays consistent for all data sets. Further, the many-to-one matching corrects some undesired cluster splits. Matching successive time steps instead of to the last makes immediate changes visible. However, errors would propagate over time leading to a loss of overall context. Further, the approach relies on t-SNE's ability to project adjacent HD points close to each other most of the time. Sufficient variation in t-SNE projections and k-Means clustering is needed to extract structures of arbitrary shape. Alternatively to clustering voxels, one could cluster directly in the parameter space using subspace clustering methods. Optimally, those algorithms find clusters in all parameter-dimension combinations. Analyzing, comparing and matching those clusters would be a challenging task.

With the proposed method we were able to gain first insights into the parameter-value distributions of a time-dependent cloud ensemble data set. Cluster ensemble techniques on k-Means clustered t-SNE plots proved to be a valid way for extracting structures from that data set, which could be found in other time steps as well using a CDF based distance measure. While the clouds are growing over time, apart from outliers and the hail quantity, their main value distributions do not change significantly. Correlations of initial parameters with the distances to the median member were found. A more detailed analysis based on different reference members and revealing higher-dimensional dependencies to initial parameters is planned for future work. Further, the application of the workflow on other data sets could lead to interesting insights.

Acknowledgements

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

[ASE15] Athawale, T., Sakhaee, E., and Entezari, A. "Isosurface visualization of data with nonparametric models for uncertainty". *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), 777–786 3.

[BM10] Bruckner, S. and Moller, T. "Result-driven exploration of simulation parameter spaces for visual effects design". *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), 1468–1476 3.

[BvLBS11] Bremm, S., von Landesberger, T., Bernard, J., and Schreck, T. "Assisted Descriptor Selection Based on Visual Comparative Data Analysis". *Computer Graphics Forum* 30.3 (2011), 891–900 3.

[DDW14] Demir, I., Dick, C., and Westermann, R. "Multi-Charts for Comparative 3D Ensemble Visualization". *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014) 3.

[DJW16] Demir, I., Jarema, M., and Westermann, R. "Visualizing the Central Tendency of Ensembles of Shapes". *SIGGRAPH Asia 2016 Symposium on Visualization*. SA '16. ACM, 2016 3.

[DS15] Dutta, S. and Shen, H.-W. "Distribution driven extraction and tracking of features for time-varying data analysis". *IEEE transactions on visualization and computer graphics* 22.1 (2015), 837–846 3.

[EKS*96] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." *Kdd*. Vol. 96. 34. 1996, 226–231 2, 4.

[FBW16] Ferstl, F., Bürger, K., and Westermann, R. "Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), 767–776 3.

[FJ05] Fred, A. L. N. and Jain, A. K. "Combining multiple clusterings using evidence accumulation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.6 (June 2005), 835–850 3, 4.

[FKRW16] Ferstl, F., Kanzler, M., Rautenhaus, M., and Westermann, R. "Visual Analysis of Spatial Variability and Global Correlations in Ensembles of Iso-Contours". *Computer Graphics Forum* 35.3 (2016), 221–230 3.

[FKRW17] Ferstl, F., Kanzler, M., Rautenhaus, M., and Westermann, R. "Time-hierarchical Clustering and Visualization of Weather Forecast Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), 831–840 3.

[FT74] Friedman, J. H. and Tukey, J. W. "A Projection Pursuit Algorithm for Exploratory Data Analysis". *IEEE Transactions on Computers* C-23.9 (Sept. 1974), 881–890 3.

[HA85] Hubert, L. and Arabie, P. "Comparing partitions". *Journal of classification* 2.1 (1985), 193–218 3.

[HBS17] Hazarika, S., Biswas, A., and Shen, H.-W. "Uncertainty visualization using copula-based analysis in mixed distribution models". *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), 934–943 3.

[HDSC19] Hazarika, S., Dutta, S., Shen, H., and Chen, J. "CoDDA: A Flexible Copula-based Distribution Driven Analysis Framework for Large-Scale Multivariate Data". *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), 1214–1224 3, 5.

[HOGJ13] Hummel, M., Obermaier, H., Garth, C., and Joy, K. I. "Comparative visual analysis of Lagrangian transport in CFD ensembles". *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), 2743–2752 3.

[JDKW15] Jarema, M., Demir, I., Kehrer, J., and Westermann, R. "Comparative visual analysis of vector field ensembles". *IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2015, 81–88 3.

[JHB*17] Jäckle, D., Hund, M., Behrisch, M., et al. "Pattern Trails : Visual Analysis of Pattern Transitions in Subspaces". *IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2017, 1–12 3.

[KH13] Kehrer, J. and Hauser, H. "Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey". *IEEE Transactions on Visualization and Computer Graphics* 19.3 (Mar. 2013), 495–513 2.

[KRRW19] Kumpf, A., Rautenhaus, M., Riemer, M., and Westermann, R. "Visual Analysis of the Temporal Evolution of Ensemble Forecast Sensitivities". *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), 98–108 4, 5.

[KTB*18] Kumpf, A., Tost, B., Baumgart, M., et al. "Visualizing Confidence in Cluster-based Ensemble Weather Forecast Analyses". *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), 109–119 3.

[Kuh55] Kuhn, H. W. "The Hungarian method for the assignment problem". *Naval research logistics quarterly* 2.1-2 (1955), 83–97 5, 6.

[KW78] Kruskal, J. B. and Wish, M. *Multidimensional scaling*. Vol. 11. Sage, 1978 4.

[LLBP12] Liu, S., Levine, J. A., Bremer, P.-T., and Pascucci, V. "Gaussian mixture model based volume visualization". *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE. 2012, 73–77 3.

[LMW*16] Liu, S., Maljovec, D., Wang, B., et al. "Visualizing high-dimensional data: Advances in the past decade". *IEEE Transactions on Visualization and Computer Graphics* 23.3 (2016), 1249–1268 2.

[LPK05] Love, A. L., Pang, A., and Kao, D. L. "Visualizing spatial multivalue data". *IEEE Computer Graphics and Applications* 25.3 (2005), 69–79 3.

[LSP*10] Lex, A., Streit, M., Partl, C., et al. "Comparative Analysis of Multidimensional, Quantitative Data". *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), 1027–1035 3.

[LT16] Lehmann, D. J. and Theisel, H. "Optimal Sets of Projections of High-Dimensional Data". *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), 609–618 3.

[MH08] Maaten, L. v. d. and Hinton, G. "Visualizing data using t-SNE". *Journal of Machine Learning Research* 9 (2008), 2579–2605 2.

[MWK14] Mirzargar, M., Whitaker, R. T., and Kirby, R. M. "Curve boxplot: Generalization of boxplot for ensembles of curves". *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), 2654–2663 3.

[OLK*14] Oeltze, S., Lehmann, D. J., Kuhn, A., et al. "Blood flow clustering and applications in virtual stenting of intracranial aneurysms". *IEEE Transactions on Visualization and Computer Graphics* 20.5 (2014), 686–701 3.

[PH13] Pöthkow, K. and Hege, H.-C. "Nonparametric models for uncertainty visualization". *Computer Graphics Forum*. Vol. 32. 3pt2. Wiley Online Library. 2013, 131–140 3.

[PHL*16] Pezzotti, N., Höllt, T., Lelieveldt, B., et al. "Hierarchical Stochastic Neighbor Embedding". *Computer Graphics Forum* 35.3 (2016), 21–30 4.

[SG02] Strehl, A. and Ghosh, J. "Cluster ensembles—a knowledge reuse framework for combining multiple partitions". *Journal of machine learning research* 3.Dec (2002), 583–617 3.

[VR11] Vega-Pons, S. and Ruiz-Shulcloper, J. "A survey of clustering ensemble algorithms". *International Journal of Pattern Recognition and Artificial Intelligence* 25.03 (2011), 337–372 3.

[WBJ*18] Wellmann, C., Barrett, A., Johnson, J., et al. "Using Emulators to Understand the Sensitivity of Deep Convective Clouds and Hail to Environmental Conditions". *Journal of Advances in Modeling Earth Systems* 10.12 (2018), 3103–3122 2, 3.

[WCR*18] Wenskovitch, J., Crandell, I., Ramakrishnan, N., et al. "Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics". *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), 131–141 3.

[WLW*17] Wang, K.-C., Lu, K., Wei, T.-H., et al. "Statistical visualization and analysis of large data using a value-based spatial distribution". *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE. 2017, 161–170 3.

[WMK13] Whitaker, R. T., Mirzargar, M., and Kirby, R. M. "Contour Boxplots: A Method for Characterizing Uncertainty in Feature Sets from Simulation Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), 2713–2722 3.