Interactive Visual Analysis of Heterogeneous Scientific Data across an Interface

Johannes Kehrer, *Student Member, IEEE*, Philipp Muigg,
Helmut Doleisch, *Member, IEEE*, and Helwig Hauser, *Member, IEEE*

Abstract—We present a systematic approach to the interactive visual analysis of heterogeneous scientific data. The data consists of two interrelated parts given on spatial grids over time (e.g., atmosphere and ocean part from a coupled climate model). By integrating both data parts in a framework of coordinated multiple views (with linking and brushing), the joint investigation of features across the data parts is enabled. An interface is constructed between the data parts that specifies (a) which grid cells in one part are related to grid cells in the other part, and vice versa, (b) how selections (in terms of feature extraction via brushing) are transferred between the two parts, and (c) how an update mechanism keeps the feature specification in both data parts consistent during the analysis. We also propose strategies for visual analysis that result in an iterative refinement of features specified across both data parts. Our approach is demonstrated in the context of a complex simulation of fluid–structure interaction and a multi-run climate simulation.

Index Terms—Interactive visual analysis, heterogeneous scientific data, coordinated multiple views.

1 Introduction

Computational simulation is used in science and engineering to investigate dynamic processes and complex phenomena. Interactive visual analysis enables the user to explore and analyze data in a guided human–computer dialog. Using proven interaction schemes such as linking and brushing, a powerful information drill-down process is supported [1]. Visual analysis is based on concepts such as coordinated multiple views, interactive feature specification via brushing, focus+context visualization, and on-demand data derivation [2].

Scientific data in a traditional application scenario is usually given in a coherent form. It can be considered, to a certain degree, as a table with rows and columns that contains multiple data attributes (given in relation to space and time). We call this a *single-part* scenario. In practice, however, we increasingly often find model and data scenarios that are more heterogeneous. They consist of two or more individual data parts that are related to each other. The data parts are, for example, computed with different simulation models, given on various data grids, with different dimensionality (e.g., 2D/3D data). Such *multi-part* scenarios present us with the challenge of integrating multiple data parts in the analysis.

Dynamic flow, for instance, is traditionally simulated with a rigid boundary. In modern *fluid–structure interactions* (FSIs), however, a movable or deformable structure interacts with an internal or surrounding fluid flow. These simulations are becoming more popular and belong, with respect to both modeling and computational issues, to the most challenging of multi-physics problems [3]. Fluid and solid parts are usually modeled individually on spatially adjoining grids that are connected by a so-called *interface*[1]. The latter represents the physical boundary between the two parts and enables them to influence each other during the simulation (compare to airplane wings or turbine blades that are deformed by the surrounding flow). Also in the climate system, as another multi-part scenario, atmosphere, ocean, ice, and land interact with each other. Ocean and atmosphere, for example, interact by means of thermal absorption, precipitation, and evaporation [4]. To understand such dynamic processes, the climate components are usually modeled individually and then coupled in the simulation, often with additional coupler modules.

Creating a coherent visualization from heterogeneous data that consists of two parts (e.g., atmosphere and ocean, or fluid and structure) is a challenge for visual analysis. How can we investigate feedback between the two data parts? The analyst is, for example, interested in areas of an ocean model that are influenced by adjacent hot areas in the atmosphere. The corresponding regions are first selected in the atmosphere via brushing. This feature then needs to be transferred to the ocean part where it can be related to ocean features and further analyzed. In our analysis framework, we realize this feature transfer by an interface that connects the two

- *J. Kehrer and H. Hauser are with Deptartment of Informatics, University of Bergen, Norway. E-mail: {johannes.kehrer, helwig.hauser}@uib.no.*
- *P. Muigg is with SimVis GmbH and Inst. of Computer Graphics and Algorithms, Vienna Univ. of Technology, Austria. Email: muigg@simvis.at.*
- *H. Doleisch is with SimVis GmbH, Vienna, Austria. Email: doleisch@simvis.at.*

Digital Object Identifier no. 10.1109/TVCG.2010.111.

[1] The term interface is used in many disciplines such as chemistry, physics, biology, or computer science. According to the Oxford English dictionary, it denotes "a point where two things meet and interact", e.g., the surface that connects two physical materials, a biological cell and another material, or a human and a computer (user interface).

data parts similar to a fluid–structure interaction. Our interface is designed such that the data parts can be given on different grids (e.g., 2D / 3D, unstructured, hybrid), with different resolutions or time-scales.

Another example that can be considered a multi-part scenario is hierarchically organized scientific data. A data part with higher data dimensionality can be related to a part with lower dimensionality, and vice versa. Multi-dimensional scientific data signifies that different attributes (e.g., temperature, pressure) are measured or simulated with respect to an m-dimensional data domain. The domain (i.e., the independent data dimensions) can be 2D or 3D space, time, but also input parameters to a simulation model. In climate research or engineering, for instance, so-called *multi-run* simulations have become an important approach to assess simulation models [5], [4]. They are used to evaluate the variability of a model and to better understand how sensitively the model reacts to its input parameters (sensitivity analysis [6]). The values of certain input parameters are varied. Simulation outputs (runs) are then computed for many combinations of the parameters. This leads to multi-run data where a collection of values exists per space / time location [7] (one value for each run).

The analysis of such higher-dimensional scientific data is generally challenging. A natural attempt in such a situation is to reduce the data dimensionality, for instance, by computing statistical aggregations along selected independent dimensions (e.g., averaging with respect to a spatial axis, the time axis, or the input parameters of the simulation). In practice, often only the aggregated data is further analyzed.

In this paper, we demonstrate that it is useful to integrate both the original multi-run data and the aggregated data part (with lower dimensionality) into the visual analysis. Similar to the simulation of a fluid–structure interaction, we construct an interface as a bridge between the two data parts. During the visual analysis, the interface is used to transfer selections (features specified via brushing) between the parts. Thus, complex relations can be investigated within and across the two data parts.

Corresponding to the multi-part scenarios described above, we have researched this problem and present the following contributions with this paper:

- We propose the construction of an interface that enables the joint visual analysis of heterogeneous scientific data that consists of two data parts.
- We propose strategies for visual analysis where the analyst works with both data parts simultaneously.
- We demonstrate the usefulness of our approach in the context of a fluid–structure interaction and a multi-run climate simulation.

2 Related Work

The integration of abstract data from multiple sources is common in *information visualization* (e.g., in relational databases [8], or web data [9]). North et al. [8] propose flexible visualization schemas built upon the snap-together visualization model, which enable the user to create multiple-view visualizations analogous to relational data schemas. Polaris / Tableau [10] supports the exploration of data cubes, where data is given at different hierarchical levels. These approaches deal with heterogeneous abstract data. In this paper, we present a visual analysis approach for heterogeneous scientific data usually given on grids over time. Cross-filtered views [11] allow interactive drill-down into relationships between multiple data attributes, also across multiple data sets. Brushing filters between pairs of views can be enabled / disabled. Cross-filtered views are neutral with respect to the data dimensionality and also support the derivation of new data attributes. With our approach, we account for the heterogeneity of the independent dimensions of space and time, similar to scenarios with multi-run data. Features can also be transferred between non-overlapping data parts such as spatially adjoining physical materials or interacting climate components. While the data is filtered with cross-filtered views, our approach leads to a joint focus–context discrimination that is related across heterogeneous data parts.

The area of *coordinated multiple views* has been steadily developing over the past fifteen years (see Roberts [12] for an overview). XmdvTool [13] allows the analysis of complex relations in multi-variate data using combinations of brushes in multiple views. SimVis [14] and WEAVE [15] are just two examples that realize the concept of a visual analysis framework for scientific data. Multiple linked views are used to simultaneously show, explore, and analyze different aspects of multi-variate data. The views are used next to each other and include 3D views of volumetric data (grids, also over time), but also attribute views such as 2D scatterplots, function graph views, or histograms. Interesting subsets of the data are interactively selected (brushed) directly on the screen, the relations are investigated in other linked views (compare also to the XmdvTool [13]).

In some systems, the result of a smooth brushing operation [16] is reintegrated within the data in the form of a synthetic *degree-of-interest* data attribute $DOI_j \in [0, 1]$ for every data item j (compare to the DOI attribution for generalized fisheye views by Furnas [17]). This data attribution represent the first interpretation level, ranging from data to knowledge [18]. Logical combinations of brushes in multiple linked views enable the specification of complex features in a hierarchical feature definition language [14]. The DOI attribution is used in all linked views to visually discriminate interesting features from the rest of the data in a focus+context visualization style [19]. Our framework is based on these concepts, extending the analysis capabilities to scenarios with heterogeneous scientific data. We connect the two data parts by an interface that transfers fractional DOI information between the parts. Complex features can be specified via (smooth) brushing within and across the data parts.

According to Fuchs and Hauser [20], scientific data stemming from different modalities (e.g., different simulation models, or measurements) can be fused at different levels in the visualization pipeline. In multi-block flow visualization, for instance, simulations are performed on multiple grid types with different resolutions [21]. Since the blocks do not represent different physical materials, a feature transfer across the blocks would not make sense. In the visualization, the blocks are usually fused at the data level (e.g., by constructing one hybrid or unstructured grid). In VisIt, for instance, data from different meshes are evaluated onto a common mesh (cross-mesh field evaluation [22]). Since the data is fused at the data level, it can be considered as a single-part scenario according to our terminology. Treinish [23] proposes a uniform data model that adjusts to the data structure and how the data is processed. Using such a data- / model-centric approach, data from different sources can be fused (or correlated), thus avoiding unnecessary interpolation or resampling to a common mesh. With our approach, fusion is performed at the feature / interpretation level [18] instead of the data level.

The treatment of *multi-run data* is rather new to the visualization community [7]. Information visualization techniques (e.g., parallel coordinates, scatterplot matrices) are used in combination with statistics to improve the understanding of the model output from multi-run simulations [24]. Nocke et al. [25] propose a system of coordinated multiple views to analyze a large number of tested model parameters and simulation runs. Statistical aggregations of the multi-run data are visualized, e.g., using linked scatterplots, graphical tables, or parallel coordinates. In their approach, however, the data is given in a coherent data part. Potter et al. [26] propose a framework that consists of overview and statistical visualizations for analyzing multi-run data. Matković et al. [5] visualize multi-run data as families of data surfaces with respect to pairs of independent data dimensions. Projections and aggregations of the data surfaces are analyzed at different levels (e.g., a 1D profile or single value per surface). In our work, we propose a more general interface concept that connects data items between two parts of scientific data and supports the transfer of fractional DOI information. This approach can also be used for multi-run data. In recent work [27], we have integrated traditional and robust estimates of statistical moments in the visual analysis of such data, where we also utilize the interface described here.

Kao et al. [28] visualize distributions over 2D multi-run data, where the distribution can apparently be represented by statistical parameters. For other cases, they propose a shape descriptor approach [29] constructing a 3D volume with the probability density function (PDF) of the data as voxel values. Mathematical and procedural operators [7] are proposed to transform the distribution data into a form where existing visualization techniques can be applied (e.g., pseudocoloring, streamlines, or isosurfaces). This operator approach is very promising due to its flexibility. However, it is not integrated in a visual analysis framework that would enable the analyst to interactively specify features within the transformed data. Recently, Potter et al. [30] extend the box plot [31] to include additional statistics. The resulting summary plot depicts different characteristics of multi-run data, however, it cannot be placed in a dense manner. In our multi-run example, we use carefully designed glyphs [32] to visualize aggregated data properties in a 3D context.

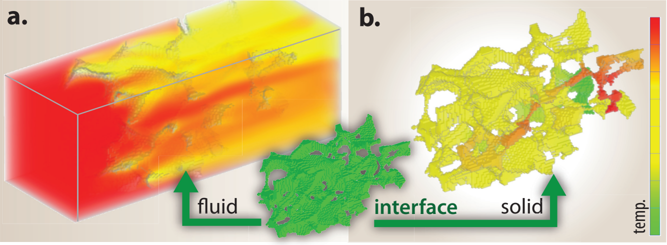

Fig. 1. The basic structure of the fluid–structure interaction: (a) simulated fluid volume with temperature mapped to color, and (b) temperature distribution in the solid part of the data. Both data parts are connected via an interface that relates cells sharing a common face between fluid and solid.

3 Sample Analysis of an FSI scenario

Fluid–structure interactions (FSIs) are complex multi-physics problems and currently an important topic in simulation research. In such scenarios, a solid structure interacts with a surrounding fluid flow, for example, by exchanging heat and / or being deformed. The variety of FSI occurrences is abundant and ranges from bridges, flexible roofs, or offshore platforms to micropumps and injection systems, from parachutes to airbags, to blood flow in arteries or artificial heart valves [3]. In the following, the study of heat transfer in an FSI scenario is used to illustrate our proposed methodology. Motivated by this example, we later come up with a more general approach that can also be applied in other scenarios with heterogeneous data such as multi-run data.

In our example, data from a multi-physics simulation of warm water flow through a cooler aluminium foam is investigated. The main goal of the domain experts is to understand how the micro structure of the simulated foam influences its thermal behavior. This knowledge can then be used to derive approximated models of the foam which can be applied within larger scale simulations. A more in-depth understanding of the flow characteristics through the simulated domain can help the application experts to experiment with different foam structures. This eventually leads to more desirable thermal properties of the foam structure.

The modeled domain contains two types of physically different materials, i.e., water and aluminium. The underlying multi-physics simulation, therefore, generates two spatially disjoint result volumes (see Fig. 1(a) and (b)). Both 3D volumes are connected by an interface which identifies common faces between fluid and solid

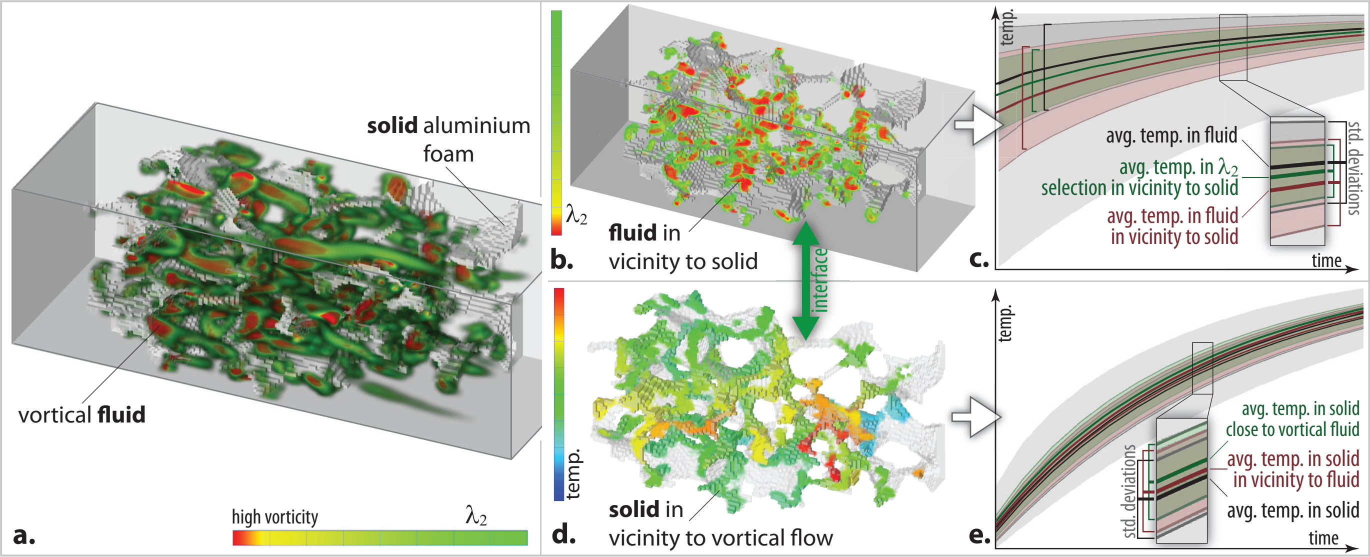
solid aluminium foam
vortical fluid
high vorticity
λ_2
a.
λ_2
fluid in vicinity to solid
b.
interface
temp.
avg. temp. in fluid
avg. temp. in λ_2 selection in vicinity to solid
avg. temp. in fluid in vicinity to solid
std. deviations
time
c.
temp.
solid in vicinity to vortical flow
d.
temp.
avg. temp. in solid close to vortical fluid
avg. temp. in solid in vicinity to fluid
avg. temp. in solid
std. deviations
time
e.

Fig. 2. Visual analysis of heat transfer using the bidirectional transfer of user-specified features: (a) vortical regions within the flow volume are selected via the λ_2 criterion [33]. Only fluid regions (b) in the vicinity of the solid and solid regions (d) in the vicinity of vortical flow are visible. In (c, e) statistical properties of selected regions are shown over time.

grid cells (illustrated in Fig. 1). During the simulation, the fluid and solid part can interact with each other via the interface, and exchange properties such as heat.

In the visual analysis, we integrate both data parts, fluid and structure. Vortices are very important in understanding flow characteristics such as heat exchange, which is the primary focus for this example. We are interested in the thermal behavior in the structure part in the vicinity of vortical flow. Since the two data parts do not spatially overlap, a selection of vortex regions in the fluid (specified via brushing) needs to be transferred to the neighboring areas in the solid part. For this purpose, we construct an interface between the data parts that is similar to the one used in the simulation. The interface is created in advance to the visual analysis, and can be saved and loaded together with the data. Grid cells that are located in the boundary region between fluid and solid are automatically correlated (the technological details are given in Sec. 4). During the visual analysis, user-specified features within these regions are instantly exchanged between the data parts via the interface. The interface can, for instance, be employed to investigate relations between flow phenomena and the resulting temperature changes within the nearby solid.

In Fig. 2 (a), vortex regions within the fluid part have been selected using the λ_2 criterion [33]. Color is mapped to the value of λ_2 with lower values, indicating stronger vortical properties, mapped as red. In Fig. 2 (b), the λ_2 selection has been restricted to fluid cells in the vicinity of the aluminium foam using the interface[2]. In order to derive quantitative properties from this selected region, the fluid temperature within it has been averaged and plotted as a green curve over time [34] (see Fig. 2 (c)). Some context is provided by plotting the overall average temperature within the fluid as a black curve and the averaged temperature in the vicinity of the solid as a brown curve (standard deviations are encoded as filled areas in the background). Since the aluminium foam is being warmed by the fluid, the averaged fluid temperature in the vicinity of the foam (brown curve) is lower than the averaged overall fluid temperature (black curve). As indicated by the green and brown curves, it is notable that the fluid temperature close to the solid is warmer when measured in regions of vortical flow.

The next step of the analysis deals with the solid portion of the simulation data. The feature transfer mechanism over the interface works bidirectionally. Thus, it is possible to project the previously defined selection of vortical flow (λ_2 criterion) onto solid regions in their vicinity. These regions are selected in Fig. 2 (d), temperature is encoded in color. The solid portions in the vicinity of vortical fluid (green curve in Fig. 2 (e)) are warmer than the average temperature in the solid (black curve) and also warmer than the remaining solid part in the vicinity of the fluid (brown curve). This is a strong indicator for a direct relation between turbulent flow around the simulated foam structure and the heating process within the structure.

4 Interactive Visual Analysis across an Interface

Motivated by the previous example of a fluid–structure interaction, our goal is to enable the joint interactive visual analysis of heterogeneous scientific data. The data consists of two parts (e.g., multi-run and aggregated data or data from a coupled climate model) that are both integrated into the visual analysis. Visual analysis is often based on the concept of user-specified interest

[2] The fully selected solid region has been transferred onto the neighboring fluid part where it is combined with the vortex feature.

per data item (resulting from feature specification via brushing). Such markups represent the first level of semantic abstraction, ranging from data to knowledge [18]. Our idea is to use a synthetic degree-of-interest (DOI) attribution [16] as a common level of data abstraction between two related parts of scientific data. In order to exchange the fractional DOI information, we construct an interface that connects individual grid cells between the data parts (similar to the fluid–structure interaction scenario). Such an abstract coordination space is also implicit in the model-view-controller pattern (see Boukhelifa and Rodgers [35], for instance).

Based on the data state reference model [36], our interface consists of the following four components illustrated in Fig. 3 and described in the following sections:

- the interface describes the *structural relation* between the two data parts (see Sec. 4.1). That is, it specifies which of the grid cells in the one data part are related to certain other cells in the other part, and vice versa. The structural relation can be generated automatically (e.g., in a pre-processing step), and is saved and loaded together with the data parts.
- during the visual analysis, the *transfer of DOI information* represents the functional aspect of the interface. It specifies how the fractional DOI information is exchanged between the data parts (see Fig. 3 (b) and Sec. 4.2). In our fluid–structure scenario, for example, a vortex feature specified in the fluid part is automatically transferred to the solid part where it can be further refined. The feature transfer works in both directions between the data parts.
- the *automatic update of feature specification* represents the dynamic aspect of the interface, which ensures consistency of the features and interactive frame rates during visual analysis. That is, the order in which the DOI information is transferred and updated between the data parts where multiple processes run in parallel (see the arrows illustrating the update process in Fig. 3 (c)).
- we also propose *strategies for visual analysis* across an interface, i.e., the interactive and iterative refinement of features that are specified within and between the two data parts (see Fig. 3 (d) and Sec. 4.4).

4.1 The Interface (structural relation)

As stated above, the interface specifies the structural relation between the individual grid cells of two parts of the scientific data (see Fig. 3). This relation needs to be generated once for a particular scenario (e.g., in an automatic pre-processing step), and can be saved and loaded. During the visual analysis, the structural relation is then used when transferring features between the data parts. In order to make the interface suitable for different scenarios with heterogeneous data, we need to consider that the two data parts can be given on various kinds of grid, with different data dimensionality, and for possibly different time steps. For all cells in one of the data parts (at a given time step), the interface stores a collection of references to all related cells (and the corresponding time step) in the other part. This allows, for instance, grid cells at a given timestep to be connected to grid cells at multiple time steps, and vice versa (e.g., when the data parts are given for different time intervals). Furthermore, a weight value is assigned to each relation between two cells. This weight determines the amount of influence a related data item has on the item in question. In the FSI scenario, for instance, it may be desirable that fluid and structure cells that are located farther apart have less influence than cells that are relatively close to each other. To make the interface as flexible as possible, the relations are separately specified in both directions. In a symmetric scenario, this can also be simplified.

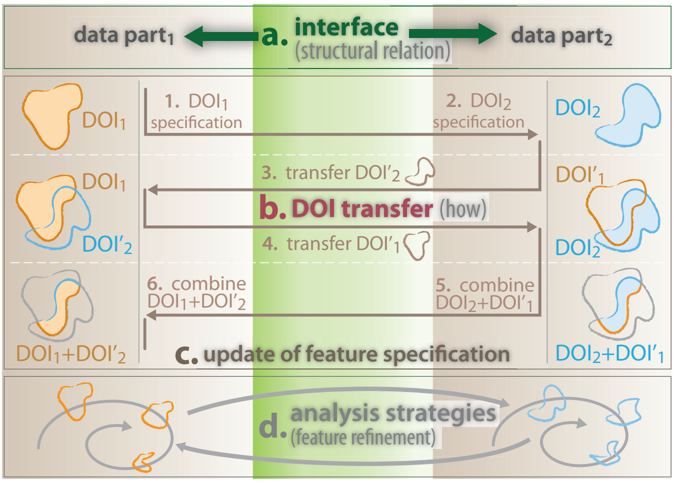

Fig. 3. In our visual analysis scenario, two parts of the scientific data are connected through an interface: the interface (a) specifies which cells in the two data parts are related to each other, (b) it specifies how the user-specified degree-of-interest (DOI) information is transferred between the data parts. Moreover, it (c) considers dynamic aspects between multiple processes to enable interaction during visual exploration, and (d) enables novel analysis strategies for iterative feature refinement.

There are three possible ways that data items can be related across different parts of the data [8]: *one-to-one*, *one-to-many*, and *many-to-many*. A one-to-one relation exists also in a traditional multi-variate dataset (single-part scenario) or when different data parts are given for the same grids / time steps. This relation is, therefore, not discussed in further detail here. In the following, we describe the many-to-many relation that exists, for instance, in a FSI simulation. The one-to-many relation is then described in the example of a multi-run scenario.

Many-to-many relation between two data parts

This kind of relation emerges, for instance, between spatially neighboring data parts such as a FSI simulation. Also in a coupled atmosphere–ocean model simulation, the two models spatially adjoin at the ocean surface and exchange properties through a coupler module (e.g., temperature, precipitation, evaporation). Since the two data parts do not spatially overlap, our approach is to

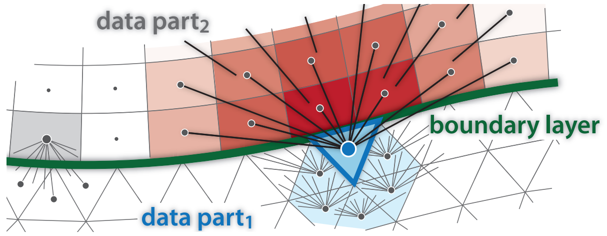

Fig. 4. Many-to-many relation between two spatially adjoining data parts: (a) a grid cell in one of the data parts can be related to multiple grid cells in the other data part, and vice versa. The weights of the grid cells related to a certain cell (blue) are encoded in red. The different data parts can represent fluid and structure, atmosphere and ocean, or fluid and fluid.

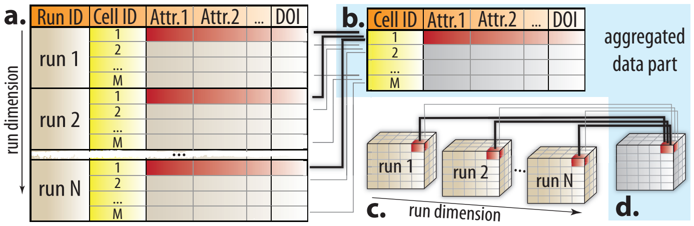

Fig. 5. One-to-many relation between two data parts with different dimensionality: every N cells in a multi-run simulation (a, c) are connected to one cell in an aggregated datablock (b, d), which share the same space/time (indicated in red).

consider the DOI transfer similar to a diffusion process of the features at the boundary between the data parts. This is in agreement, for instance, with the oceanographers' concept of the upper ocean layer that is influenced by the atmosphere (influence is decreasing with depth).

As shown in Fig. 4, the relationship between grid cells sharing a common boundary between the data parts can be translated into a many-to-many interface. The N data items that are close to the boundary layer are connected to M data items which lie in their vicinity in the second data part, and vice versa. As illustrated for the blue grid cell in Fig. 4, the influence of the related grid cells (i.e., the weight values encoded in red) decreases with the spatial distance between the cells.

An interface such as the one used in the fluid–structure interaction example can be automatically constructed as follows (see Fig 4): For every $cell_i$ in data part$_1$ that is within a certain distance $dist_{\max}$ to the boundary surface, all grid cells in data part$_2$ that are within a distance $dist_{\max}$ to $cell_i$ are added to the collection of related cells. The individual weights for the related cells are, for example, specified as a function of the distance $dist_{i,j}$ between the cells and an importance value of the cell CI_j, i.e.,

$$w_j = CI_j \frac{dist_{\max} - dist_{i,j}}{dist_{\max}} \quad .$$

CI_j is usually proportional to the actual volume of the grid cell, giving larger cells a higher influence than smaller ones. In some cases, however, the opposite may be desirable. In simulation, for instance, smaller cells are often used in regions of special interest. In such a case, smaller cells can then receive a higher importance value CI_j than larger cells.

One-to-many relation between two data parts

This kind of relation exists, for example, between data parts that are specified at two different hierarchical levels. Examples are scale space representations of scientific data where data is given at different resolutions [37] or multi-run and aggregated data that are given with different dimensionality. In the latter case, the higher dimensional data part represents the original multi-run data (with additional independent dimensions for the input parameters to the simulation). In Fig. 5 (a) and (c), a collection of N values exists for the same data attribute for every grid cell (e.g., 100 temperature values per cell for a simulation with 100 runs). To analyze the distribution of values, statistical properties such as mean or standard deviation can be computed with respect to the run dimension (or another independent data dimension). The result of this aggregation represents the second data part given at a lower dimensionality. In Fig. 5 (b) and (d), every single cell in the aggregated data part is, therefore, related to the N cells in the multi-run data that share the same space and time, and vice versa.

4.2 Transfer of Degree-of-Interest Information

The DOI transfer represents the functional aspect of the interface. It is based on the structural relation between the two data parts (see Sec. 4.1). For every data item i in one data part, the transferred DOI'_i is computed from the related data items in the other part, and vice versa. This transferred DOI information is then combined with the local one in the data part (e.g., logical AND / OR). Since the DOI transfer works bidirectionally, we need to ensure that the transferred feature is not transferred back, which would lead to inconsistencies in the feature specification (see also Sec. 4.3). We propose three different ways of transferring the DOI information: (1) weighted sum, (2) maximum (or minimum) weighted DOI value, and (3) maximum (or minimum) DOI value without weighting. Depending on the user's needs, one can switch between these options during the visual analysis. This opens up interesting opportunities for analytic procedures (see also Sec. 4.4).

With the first approach, the weighted sum of the DOI values of related cells is computed for every data item i: $DOI'_i = \frac{1}{\sum_j w_j}\left(\sum_j w_j \cdot DOI_j\right)$. For the one-to-many relation (e.g., when working with hierarchically organized data parts) this represents the transfer of the average of the related DOI values. For the many-to-many relation this kind of DOI transfer can be seen as a diffusion of the DOI information across the interface. For cases in which data is given in a continuous form, the process is similar to integrating over the weighted DOI values of the related cells. This weighted transfer is well suited for examining the degree to which the

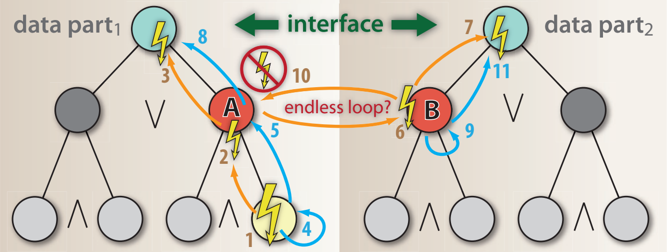

Fig. 6. Relating complex features that are specified in a hierarchical manner. Yellow arrows represent node invalidations and blue arrows represent updates.

related cells are part of the focus in the other data part (e.g., 20 out of 100 related cells are selected). However, it has the drawback that isolated DOI features are deemphasized due to averaging of the DOI values, e.g., when only one of the related cells has a maximum DOI value and all other cells are part of the context.

In order to preserve such DOI peaks, we suggest also to allow the maximum of the weighted DOI values of related cells to be transferred, i.e., $DOI'_i = \max_j(w_j \cdot DOI_j)$. As a third alternative, the user can choose to neglect the weight values, and transfer the maximum value of the related DOI information only. This can be useful, for instance, in order to preserve features even though only a few related cells have large DOI values, or relatively low weights. Examples are grid cells in a FSI scenario or multi-model simulation without considering the actual inter-cell distance or cell importance. The three methods for DOI transfer are suitable for different stages of a visual analysis, which is discussed in section 4.4.

4.3 Automatic Update of Feature Specification

In this section, we describe dynamic aspects of interlinking two data parts. During the visual analysis, the feature specification is automatically updated by multiple threads to ensure consistency and responsiveness of the application. Features can be specified by logical combinations of brushes within and across views. In our framework, the resulting DOI information within an attribute view (e.g., scatterplot, histogram) is represented as a leaf node in a hierarchical feature definition language [14]. The nodes are combined by logical AND / OR-operations in order to specify three levels of focus (see Fig. 6). The different focus levels and the context are encoded in color in every attribute view [19]. The DOI information is thus defined at every node in the feature tree and a flag indicates whether the information is currently up-to-date. As soon as the DOI information at a certain tree node becomes outdated (e.g., when altering a brush in a view), all update processes are suspended. The out-of-date event is propagated up to the tree root (see the flash symbols 1–3 in Fig. 6). Update threads are then restarted, and the feature specification is updated in a depth-first manner, starting with the deepest node in the tree that is out-of-date (steps 4, 5, and 8 in Fig. 6).

The feature trees in two data parts can be related by exchanging the DOI information of two nodes given at the same hierarchy level (e.g., nodes A and B in Fig. 6). The naïve approach is to set the related node B out-of-date after the DOI information in node A is updated (i.e., after step 5)—this is then propagated up to the tree root in data part₂ and starts the corresponding update threads. This approach works well as long as the data parts are related only in one direction. If the relation is established in both directions, node A would also be set out-of-date after node B is updated (illustrated in step 10). This would cause an endless loop of updates. To avoid this problem, we do not set node B out-of-date in step 6. Instead a synchronized update of nodes A and B is performed in step 5. Subsequently, only the parent node of B is set out-of-date (step 7).

The sequence of events as to how the related nodes A and B exchange their DOI information is illustrated by arrows in Fig 3 (c). First, the feature specification in node A (DOI_1) and node B (DOI_2) is updated, combining the DOI information of the respective child nodes. When exchanging the features via the interface, we need to ensure that the transferred DOI information is not transferred back. This would lead to inconsistencies in the feature specification. In steps 3 and 4 in Fig. 3 (c), therefore, the DOI information is first transferred between the data parts (see DOI'_1 and DOI'_2) and stored temporarily. After that, the transferred DOI can be combined with the local one (steps 5 and 6 in Fig. 3 (c)). During this process, all operations are performed by the threads of only one data part (potential updates of the feature specification in the other data part are suspended).

After nodes A and B have exchanged their DOI information as described above, only the parent node of B is set out-of-date (step 7 in Fig. 6). This restarts the update threads in the feature tree in data part₂. Since node B itself has not been set out-of-date, steps 9 and 10—leading to an endless loop—are not performed.

4.4 Strategies for Visual Analysis

Interactive visual analysis enables the user to enter a *visual dialog* with the data. The employed procedure usually follows Shneiderman's information seeking mantra [1] (overview first, zoom and filter, details-on-demand) or Keim's recent modification for visual analysis [38] (analyze first, show the important, zoom, filter and analyze further, details-on-demand). The analysis process usually takes place in a single-part scenario. When this is extended to two data parts, the pattern has to be adapted accordingly. Additional iteration loops are introduced between the data parts as illustrated in Fig. 3 (d). With spatially adjoining data parts, for example, features are iteratively specified in one data part by brushing. The relations of the features—transferred by the interface—are also inspected in the other data part, e.g., in the spatial context using a 3D view or in attribute views (compare to the FSI scenario in Sec. 3). At a certain point, the analysis moves over to the other data part, possibly also with certain iterations, before it can go back to the first data part, and so on.

We have worked through several analysis scenarios with two hierarchically related data parts (Sec. 5 describes one such analysis of multi-run climate data). From these scenarios, we see that it is useful to have views that show the data at the aggregated and detail level next to each other. The analysis usually starts at the aggregated level (overview first). Statistical properties—computed from the data part given with more detail (e.g., the multi-run data)—are investigated at this level. Interesting data characteristics can be selected such as distributions that have a high variability or contain irregularities such as outliers. While interactively brushing the aggregated properties, the collections of related data values are instantly highlighted in another view at the detail level. After several iterations at the aggregated level, the analysis continues in the data part that is given with more detail. The features can be further refined here (e.g., selecting / excluding individual data values that are outliers). The relations are again checked in both data parts, and so on.

An analysis pattern with respect to the DOI transfer is to begin with a maximum transfer first. This is independent of the quantitative influence which the related data items have on each other (e.g., the distance between cells in a FSI scenario). That is useful, for instance, not to "lose" features in cells with small weight values due to averaging. Such a maximum DOI transfer enables the analyst to look up where features co-exist in both data parts. At a certain stage of the analysis, the analyst decides to change to a weighted DOI transfer. This results in a more quantitative analysis of the relations between the data parts, i.e., the degree to which the features co-exist. With two hierarchically related data parts (one-to-many), one can investigate how many of the related cells (e.g., in the multi-run data part) are part of the focus. For spatially neighboring parts, the weighted DOI transfer also gives an indication of how close or distant the related cells are. For scenarios with FSI or coupled climate models, this transfer corresponds to the physical properties of a diffusion process.

Another important aspect of related analysis procedures is that data attributes can be transferred across the interface as well (compare to data transformations in the data state reference model [36]). Using an integrated *data calculator* module with a respective graphical user interface, additional data attributes can be derived from existing ones that are possibly located in the other data part. To do so, the structural relation between the data items in the data parts is used (see Sec. 4.1). The new attributes are thereafter available for full investigation in all linked views. We will benefit from this mechanism in the demonstration (Sec. 5), where statistical attributes are derived from multi-run data during the visual analysis.

5 Analysis of Multi-run Climate Data

The visual analysis of heterogeneous scientific data is exemplified in the context of a climate data analysis. We investigate data from a multi-run simulation of a prominent palaeoclimatic cold event. The anomaly was caused by a meltwater outburst from Lake Agassiz, an immense glacial lake located in the center of North America. About 8200 years ago, the lake drained due to climate warming and melting of the Laurentide Ice Sheet. The investigated data stems from the CLIMBER-2 coupled atmosphere–ocean–biosphere model that simulates a cooling of about 3.6 K over the North Atlantic [39].

With a sensitivity analysis, an important goal for the climate modelers is to better understand the variability of a simulation model with respect to certain model parameters. Identifying those parameters that have the most influence can help to validate the model and also guide future research efforts [6]. Multiple simulation runs are computed with varied initial parameters. In our case, two diffusivity parameters of the ocean model are altered, one horizontal (diff_h) and one vertical (diff_v), with ten variations each. This leads to a dataset with a total of 100 (10×10) runs. For each run, the data is given for 500 years on 2D sections (latitude × depth) through the Atlantic, Indian, and Pacific ocean. In the following, we present a selection of results from a visual sensitivity analysis of the ocean part of the CLIMBER-2 model based on the input parameters diff_h and diff_v.

5.1 Basic Setup for the Visual Analysis

Since the number of independent dimensions in the multi-run ocean data is already challenging (5 dimensions, i.e., a 2D section for each ocean, time, and two run parameters with 10×10 runs), a traditional visual analysis is difficult. Reducing the data dimensionality can help, for instance, by computing statistical aggregates along an independent data dimension. Such an example is to consider averages over time instead of all the individual data values. For the ocean data, we compute statistics with respect to the two run-dimensions. The aggregated data properties are reintegrated in our framework through an attribute derivation mechanism. The result is stored in a separate data part with lower dimensionality than the original data (i.e., a 2D section per ocean over time).

For the visual analysis, we connect the data part that contains the multiple runs and the aggregated data part by an interface. The interface is created automatically during the data conversion and is loaded together with the data parts at the beginning of the analysis session. As discussed in section 4.1, a one-to-many relation is established between each aggregated cell and the collection of multi-run values given for the same space and time (see Fig. 5 (c) and (d)). Brushing, for instance, an aggregated cell also selects the related distribution of values in the multi-run data (at the same timestep). Since the two data parts are connected by the interface, we can go back and forth between the original data and aggregated statistics during the visual analysis.

In the following analysis, we first familiarize ourselves with the data by means of an overview visualization (in the aggregated data part). This is based on glyphs showing derived statistical properties computed from the multi-run data. In the aggregated part, we are able to identify certain cells which contain interesting outliers (with respect to a sensitivity analysis). The selection is automatically transfered via the interface to the multi-run data part. The feature is further investigated and refined, which is also reflected back to the aggregated data part. In the analysis, the parameter settings that lead to the selected outliers can be identified.

Firstly, we want to obtain an overview of the multi-run ocean data. At every timestep, statistical properties are computed from each distribution of multi-run values per grid cell. We are, for instance, interested in distributions where the outputs from different runs have a high variation. For this purpose, we compute the quartile information that is commonly represented in box plots [31]. The three quartiles divide the collection of 100 values per grid cell—one value per run—into four equally populated parts: 25% of values are smaller than the *lower quartile* q_1, 50% are smaller/larger than the *median* q_2, and 25% of the values are larger than the *upper quartile* q_3. The median is a robust estimate of the center of a distribution (as compared to the mean) and the *interquartile range* ($IQR = q_3 - q_1$) is a more robust estimate for the standard deviation [40]. Carefully designed glyphs, placed as billboards in 3D, can be used to represent multiple properties per grid cell [32].

The glyphs provide qualitative information about the data distribution with respect to the multiple runs. In Fig. 7(a), four statistical properties are represented per aggregated cell at timestep 100: the median temperature is encoded in color[3], the interquartile range is mapped to the overall glyph size, the upper glyph shape represents the distance $q_3 - q_2$, and the lower shape shows $q_2 - q_1$. Large interquartile ranges have been brushed, opacity represents the respective DOI values. The upper and lower shape of the glyphs are based on super ellipses [32]. Each shape represents an attribute by changing from a star (small value), to a diamond, to a circle, and a box representing a large value (see the glyph legend in Fig. 7(a)). Even though the figure may contain some visual cluttering, it gives a qualitative overview about the data distribution over all runs (at the given timestep). We see a couple of interesting locations (larger glyphs) where the corresponding distribution of multi-run values have a high variation. The upper/lower glyph shapes also provide information about the skewness of the distribution. Due to its horizontal symmetry, the glyph shape can usually be mentally reconstructed when the glyph is partially occluded. The user can also zoom and rotate the visualization.

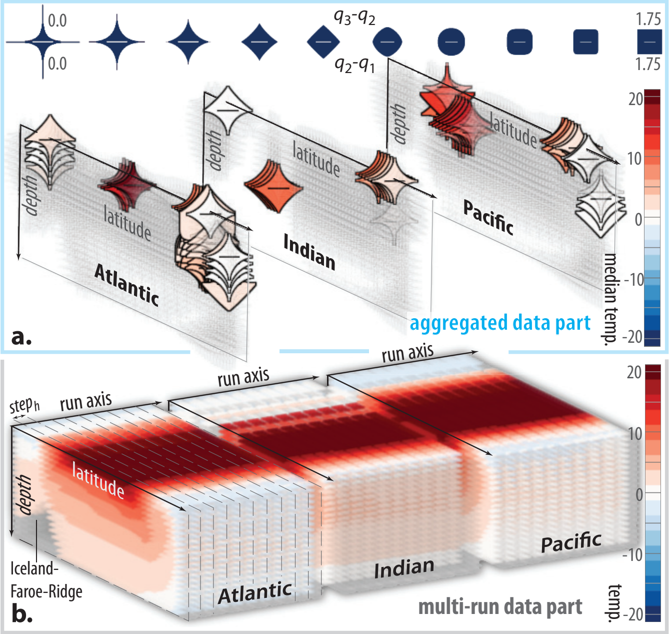

Fig. 7. Multi-run climate data at timestep 100 given for two hierarchical levels: (a) glyph-based visualization of four aggregated properties from the multi-run data (color, overal size, upper/lower glyph shape). (b) the original multi-run data on 2D cross sections through the Atlantic, Indian, and Pacific ocean. The run parameters are encoded in one of the spatial dimensions (run axis). Camera settings in both views are synchronized.

Fig. 7(b) depicts the multi-run data part at the same timestep. For each run, temperature is shown on a cross section through the Atlantic, Indian, and Pacific ocean. The 2D sections (latitude × depth) are hierarchically arranged next to each other. The two run dimensions of the data are embedded by (re-)using one of the spatial dimensions of the visualization (denoted as run axis). The location r along the run axis is determined by the input parameters to the simulations, i.e., $r = \mathit{diff}_h \cdot \mathit{step}_h + \mathit{diff}_v \cdot \mathit{step}_v$, where step_h is chosen slightly larger than $10 \cdot \mathit{step}_v$. This leaves some space between cross sections resulting from different settings for diff_h (illustrated in Fig. 7(b)). Both step sizes can be specified by the user. During interaction, the camera settings for the aggregated and multi-run view are synchronized.

5.2 Outlier analysis in the aggregated data part

As a next step, the influence of the ocean diffusivity parameters on the simulation output is investigated. We focus on grid cells that contain interesting multi-run outliers. These are values resulting from individual runs that strongly diverge from the output of other runs (for the same grid cell and timestep). Identifying such outliers can be useful for finding possible errors in the model or unsuitable settings for the model parameters. We compute additional data properties from the multi-run data using the integrated data derivation mechanism

[3] The color maps are based on the work of Brewer [41]. Discrete maps are chosen to allow more quantitative statements about the data.

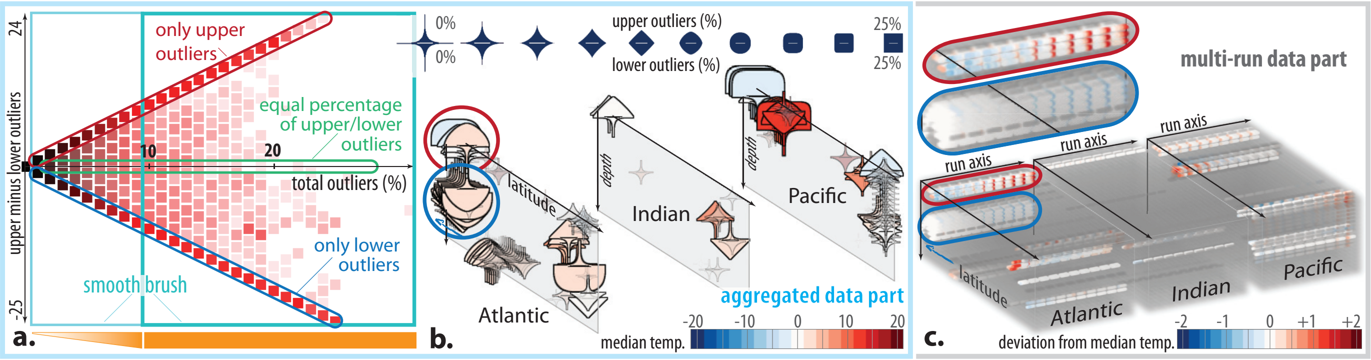

Fig. 8. Analyzing cells that contain at least 10% of outliers: (a) scatterplot showing the percentage of total outliers (x-axis), and a measure to determine how the outliers are distributed (y-axis), i.e., are more located above q_3 (upper outliers) or below q_1 (lower outliers). Aggregated outlier properties are depicted using glyphs (b), the selected cells are also shown for the multi-run data (c).

of our framework. The resulting properties are stored in the aggregated data part. We create a 2D scatterplot that can answer two questions per multi-run distribution:

- what percentage of the multi-run values given for a grid cell / distribution represent outliers (x-axis), and
- how are the outliers distributed (y-axis). That is, are more outliers located above q_3 or below q_1, are they equally distributed, etc.

Univariate measures of *outlyingness* often consider the distance of the samples to the data center, normalized by the standard deviation. Such measures can be estimated in a classical or a robust way [40]. Data values that lie more than $1.5 \times IQR$ away from the upper or lower quartile are often considered as "mild" outliers, and values that differ by more than $3 \times IQR$ are considered as "extreme" outliers [42]. At this stage of the analysis, we consider mild and extreme outliers as equally important. In Sec. 5.3, however, we treat them differently.

For each distribution of multi-run values at a timestep, we derive the percentage of *upper outliers* (% data values $\geq q_3 + 1.5 \times IQR$) and *lower outliers* (% data values $\leq q_1 - 1.5 \times IQR$). The scatterplot in Fig 8 (a) shows aggregated properties for all grid cells and timesteps[4]. The percentage of total outliers per grid cell (at a timestep) is mapped to the x-axis. A measure that expresses whether there are more upper or lower outliers is represented on the y-axis (i.e., upper minus lower outliers). In the view, the number of data items per rectangle is encoded by its luminance and the DOI values are represented by color (pure red represents a maximal DOI value). Grid cells with certain outlier characteristics can be investigated via brushing: Data items at $(0, 0)$ contain no outliers according to the chosen measure. Items along the diagonals contain either only upper or lower outliers. Items located on the x-axis ($y = 0$) contain the same number of upper and lower outliers. Using a smooth brush [16], we focus on grid cells where more than 10% of the multi-run values diverge strongly from the rest (with a transition to cells containing no outliers, illustrated as an orange gradient below Fig. 8 (a)). While brushing these aggregated characteristics, the selection is instantly transferred to the multi-run data part via the interface. The spatial relation of the feature can be investigated in Fig. 8 (b) and (c).

The glyphs in Fig. 8 (b) depict the derived outlier characteristics at timestep 60. Color represents the median temperature and the overall glyph size represents the percentage of total outliers per cell (at the timestep). The upper and lower glyph shape shows the percentage of upper and lower outliers, respectively. In Fig. 8 (c), the corresponding deviation of multi-run values from the median temperature is visualized. A group of cells with mainly upper outliers (round upper glyph shape) is visible in the north of the Atlantic (see red ellipses in Fig. 8 (b) and (c)). Another group of cells with many lower outliers is located north of the Iceland-Faroe-Ridge in the Atlantic (see the blue ellipses). By changing the depicted timestep, one can observe that the feature with lower outliers propagates northwards and downwards near the seabed over time. The feature also extends over the north pole to the other parts of the arctic sea (not visible at this timestep). At a later stage of the simulation, an increasing number of runs results in such lower (cooler) outliers compared to the rest (blue ellipses in Fig. 8 (b) and (c)). We further investigate this feature.

We focus on cells that contain more lower than upper outliers. To allow such a relative selection, the data mapped to the y-axis in Fig. 8 (a) is normalized. The respective data attribute (upper minus lower outliers) is, therefore, divided by the corresponding percentage of total outliers (x-axis). The resulting scatterplot is shown in Fig. 9. For each column of total outliers (x-axis), the combinations of upper and lower outliers are now equally distributed on the vertical axis (this is illustrated for the example of 12% total outliers in Fig. 9). Accordingly, it is now possible to brush the ratio between upper and lower outliers. Data items that (1) contain at least 10% of outliers at a timestep (x-axis), and that (2) have at least 75% lower outliers—compared to the percentage of upper outliers—are in full focus (see also the smooth

[4] Since the point size in this plot has been increased, it is similar to a 2D histogram using colored rectangles to represent the bar height.

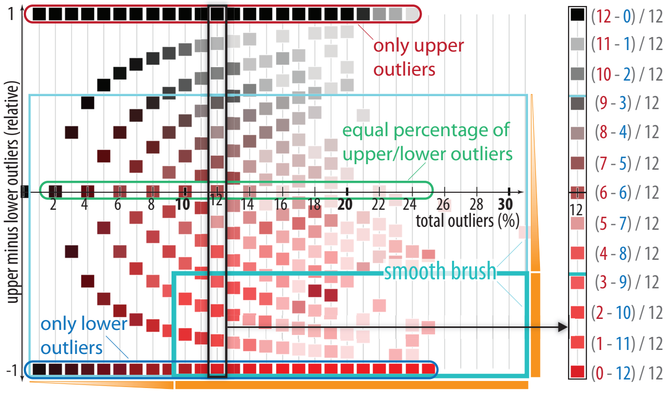

Fig. 9. Distributing the data items from Fig. 8 (a) uniformly on the vertical axis supports brushing of certain outlier characteristics. Grid cells are selected that contain at least 10% of outliers of which at least 75% are lower outliers. For the example of 12% total outliers, the possible distributions of upper minus lower outliers (red and blue number) is shown.

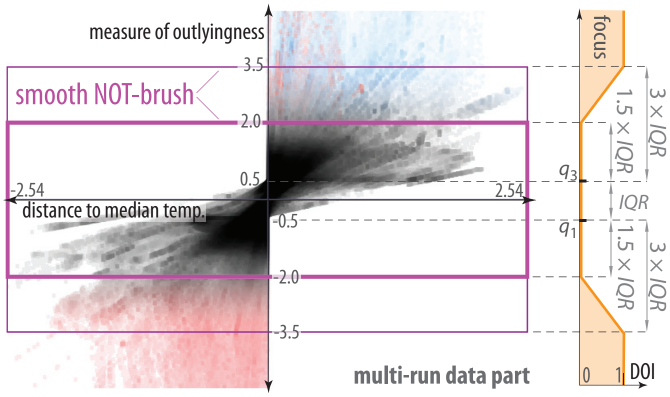

Fig. 10. Outliers are brushed using derived attributes in the multi-run data part: mild outliers are vertically located above or below ± 2.0 and extreme outliers are located above or below ± 3.5. Features selected in multiple views are highlighted in red (focus), features only selected in the current view are depicted in blue, and context information in black.

extension of the brush where the DOI linearly decreases, illustrated as orange gradients in Fig. 9). The respective feature is further analyzed in the following section.

5.3 Outlier analysis in the multi-run data part

Up to now, our analysis was mainly based on aggregated properties. Since both data parts are connected through an interface, we can go back to the original multi-run data and further refine our selection of lower outliers. In the following, the model sensitivity with respect to the input parameters, $diff_h$ and $diff_v$, is investigated for the specified feature. Our goal is to identify (1) the grid cells with the specified outlier characteristics, and (2) the parameter settings that result in such outliers. A measure of outlyingness is thus derived in the multi-run data part, which also allows us to differentiate between mild and extreme outliers. For each multi-run value x_j, the deviation from the center of the corresponding distribution is normalized by the interquartile range, i.e., $\frac{x_j-(q_1+q_3)/2}{IQR}$. Values inside $[q_1, q_3]$ are thereby mapped to the interval $[-0.5, 0.5]$. Note that the median does not have to be zero on this scale.

The scatterplot in Fig. 10 shows the described measure of outlyingness for the multi-run data (y-axis), and the corresponding deviation from the median temperature per distribution (x-axis). We brush multi-run values that represent extreme outliers with a smooth transition to mild outliers (see the illustration on the right of Fig. 10). In the scatterplot, such extreme outliers are vertically located above or below ± 3.5 and deviate by more than $3 \times IQR$ from the upper or lower quartile, respectively. Mild outliers are located above or below ± 2.0. In Fig. 10, different levels of focus+context [19] are discriminated in color: the context is shown in black, data items only selected in the local view are encoded in blue, and items selected in both data parts are highlighted in red. Since the interface works bidirectionally, the maximum DOI value per multi-run distribution is also transferred to the related grid cell in the aggregated data. Aggregated cells where the related distribution contains only mild outliers accordingly receive a low DOI value.

As a next step, multi-run values that are relatively similar to the median temperature of the corresponding distribution are excluded from the selection (with a smooth brush on the x-axis in Fig. 10, not shown here). This is to account for distributions with a very small interquartile range, where the chosen measure of outlyingness becomes less significant (as compared to larger interquartile ranges).

In the following, we investigate the temporal evolution of the previously specified feature of lower outliers (aggregated data part) that has been refined in the multi-run data part to also identify the parameter settings causing these outliers. Fig. 11 shows the aggregated and multi-run data at three different timesteps, represented as columns. The aggregated outlier properties are visualized in the top row. A diverging color map is used for the multi-run data (middle row) to encode the deviation from the median temperature per grid cell. A view from above (bottom row) is used to identify the input parameter settings resulting in the selected outliers ($diff_v$ settings are also color-coded).

At timestep 60, the selected multi-run values strongly deviating from the rest of the distribution are visible north of the Iceland-Faroe-Ridge (see the blue ellipses in Fig. 11 (b) and (c)). Since the corresponding $diff_v$ and $diff_h$ settings are spatially encoded in the visualization, we can see that these outliers mainly result from larger $diff_v$ settings[5] (see the inset in Fig 11 (c)). At this stage of the simulation, multi-run values that are also simulated with larger values for the horizontal ocean diffusivity ($diff_h$) deviate earlier from the other runs. For these

[5] These runs are located on the right side each, because $diff_v$ input values are encoded with a smaller step size than $diff_h$ values.

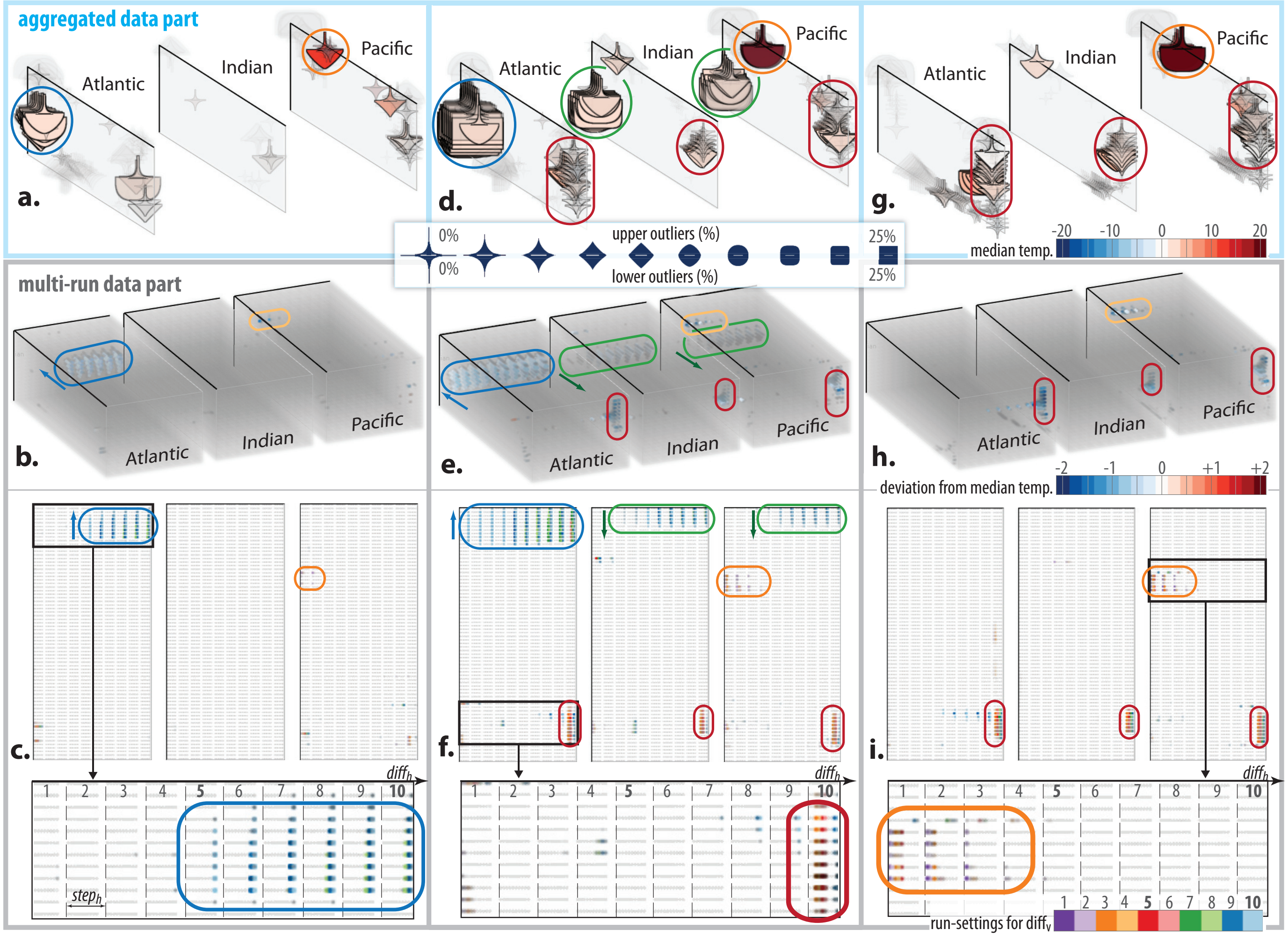

Fig. 11. Investigation of the specified feature (multi-run and aggregated data) at timestep 60 (left), timestep 120 (middle), and timestep 250 (right). In the top row, four derived properties are visualized (median temp., % upper/lower outliers, % total outliers). The individual runs that result in outliers are visible in the 3D context (second row), and in a view from above (last row). Here, the two run parameters are embedded by (re-)using one of the spatial dimensions in the visualization ($diff_v$ is also color coded).

grid cells and run settings, the model changes from its standard behavior to another climate condition.

At year 120 the feature of lower outliers (blue ellipse in the north of the Atlantic) has also propagated to the other parts of the arctic sea (green ellipses in the Indian and Pacific ocean). After extending over the north pole, it is propagating southwards (indicated with arrows). It is still only larger settings for $diff_v$ that produce these outliers (see upper part of Fig 11 (f)). On the other hand, a condition has established in the southern region in all three oceans where a few runs constantly result in different output values to the rest (see red ellipses). These outliers result from large $diff_h$ settings and are, therefore, represented to the right of each ocean. A similar behavior is also visible at year 250 (see the red ellipses in Fig 11 (h) and (i)). At this stage of the simulation, the outliers previously visible in the north have already disappeared. A condition has been established in the north where the runs mainly result in similar outputs.

Over the investigated timespan, certain runs in the northern Pacific also produce a larger number of lower outliers (see the orange ellipses in Fig. 11). These outliers result mainly from smaller settings for $diff_h$ and $diff_v$ (e.g., see the inset in Fig. 11 (i)). As a next step, we change our selection in the aggregated data part to select grid cells that contain more upper than lower outliers. A similar analysis is performed, where the parameter settings producing these upper outliers are investigated. Due to space limitations, this is not shown here.

In summary, we performed a visual sensitivity analysis of a multi-run climate simulation. In our analysis framework, multi-run and aggregated data were integrated and related by an interface, which supports the investigation of features across both data parts. Statistical properties were computed from the distributions of multi-run values. Based on these properties, interesting outlier characteristics could be brushed in the aggregated data part. The feature was automatically transferred to the multi-run data via the interface where it was further investigated. Individual runs that substantially deviate from the other values of the distribution could be identified together with the corresponding input parameter settings. By connecting both data parts via the interface, the analyst can go back and forth between multi-run and aggregated data, which enables a powerful analysis.

6 Conclusion and Future Work

The joint visual analysis and exploration of heterogeneous scientific data is a crucial and challenging task. In this paper, we propose a systematic approach to the interactive visual analysis of two heterogeneous parts of scientific data. Analogous to the related simulation scenarios, we construct an interface between the data parts which connects data items in the one part to data items in the other, and vice versa. We propose different ways of how a user-specified degree-of-interest attribution can be transferred between the data parts. Instead of performing fusion between the parts at the data level—this is often not practical in scenarios including multi-run simulation data or fluid–structure interactions—we perform the fusion on the first semantic / interpretation level explicitly represented as user-specified features [18]. Our approach is demonstrated in two visual analysis scenarios with heterogeneous scientific data, which were conducted in collaboration with domain researchers.

For data parts specified at hierarchically different levels, the integration of derived statistical attributes in the analysis process has shown great potential. It enables the analyst to work simultaneously in both—the data part containing the actual data, and the aggregated data part representing summary information. The analyst can go back and forth in an iterative manner, analyzing the data at different hierarchical levels. Relations between these data parts can thereby be identified through the visualization and iteratively refined. Such a tight integration of a computational and interactive analysis methodology agrees well with the requirements for prototypic visual analytics solutions [38].

In future work we will focus on extending our approach to scenarios with multiple data parts (e.g., given at multiple aggregated levels). We also aim at further integrating statistical properties, yielding quantitative results into our visual analysis framework. Here again, we want to show how visual analysis and statistics can interact in a feedback loop to gain in-depth insight into the data. We also want to identify further analytical patterns that involve our interface.

Acknowledgments

The authors thank Thomas Nocke, Michael Flechsig, and colleagues from the Potsdam Institute for Climate Impact Research, Germany, for fruitful discussions, valuable comments on this paper, and for providing the climate simulation data. We thank Andreas Lie for implementing the glyph renderer and Stian Eikeland for the data conversion (both from the Univ. of Bergen). We also thank Matthew Parker (Univ. of Bergen), Brendan McNulty (Allegro Language Service, Bergen), David Horn, and our anonymous reviewers for their valuable comments that helped to improve this paper. The CFD data is courtesy of Innovative Computational Engineering GmbH (www.ice-sf.at), Leoben, Austria. This work was supported in part by the Austrian Research Funding Agency (FFG) in the scope of the projects "AutARG" (No. 819352) and "PolyMulVis" (No. 823855).

References

[1] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proc. IEEE Symp. Visual Languages*, 1996, pp. 336–343.

[2] H. Hauser, "Interactive visual analysis – an opportunity for industrial simulation," in *Simulation and Visualization*, 2006, pp. 1–6.

[3] H.-J. Bungartz and M. Schäfer, Eds., *Fluid-Structure Interaction – Modelling, Simulation, Optimisation*, ser. Lecture Notes in Computational Science and Engineering. Springer, 2006, vol. 53.

[4] B. Hibbard, M. Böttinger, M. Schultz, and J. Biercamp, "Visualization in Earth System Science," *SIGGRAPH Comput. Graph.*, vol. 36, no. 4, pp. 5–9, 2002.

[5] K. Matković, D. Gračanin, B. Klarin, and H. Hauser, "Interactive visual analysis of complex scientific data as families of data surfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1351–1358, 2009.

[6] D. M. Hamby, "A review of techniques for parameter sensitivity analysis of environmental models," *J. Environmental Monitoring & Assessment*, vol. 32, no. 2, pp. 135–154, 2004.

[7] A. Love, A. Pang, and D. Kao, "Visualizing spatial multivalue data," *IEEE Comput. Graph. Appl.*, vol. 25, no. 3, pp. 69–79, 2005.

[8] C. North, N. Conklin, K. Indukuri, and V. Saini, "Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table databases," *Information Visualization*, vol. 1, no. 3-4, pp. 211–228, 2002.

[9] M. Cammarano *et al.*, "Visualization of heterogeneous data," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1200–1207, 2007.

[10] C. Stolte, D. Tang, and P. Hanrahan, "Polaris: A system for query, analysis, and visualization of multidimensional relational databases," *IEEE Trans. Vis. Comput. Graph.*, vol. 8, no. 1, pp. 52–65, 2002.

[11] C. Weaver, "Cross-filtered views for multidimensional visual analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 2, pp. 192–204, 2010.

[12] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *Proc. Coordinated & Multiple Views in Exploratory Visualization*, 2007, pp. 61–71.

[13] M. Ward, "XmdvTool: Integrating multiple methods for visualizing multivariate data," in *Proc. IEEE Visualization*, 1994, pp. 326–336.

[14] H. Doleisch, M. Gasser, and H. Hauser, "Interactive feature specification for focus+context visualization of complex simulation data," in *Proc. Symp. on Visualization (VisSym)*, 2003, pp. 239–248.

[15] D. Gresh *et al.*, "WEAVE: a system for visually linking 3-D and statistical visualizations applied to cardiac simulation and measurement data," in *Proc. IEEE Visualization*, 2000, pp. 489–492.

[16] H. Doleisch and H. Hauser, "Smooth brushing for focus+context visualization of simulation data in 3D," *Journal of WSCG*, vol. 10, no. 1, pp. 147–154, 2002.

[17] G. W. Furnas, "Generalized fisheye views," in *Proc. Human Factors in Computing Systems (CHI '86)*, 1986, pp. 16–23.

[18] M. Chen *et al.*, "Data, information, and knowledge in visualization," *IEEE Comput. Graph. Appl.*, vol. 29, pp. 12–19, 2009.

[19] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser, "A Four-level Focus+Context Approach to Interactive Visual Analysis of Temporal Features in Large Scientific Data," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 775–782, 2008.

[20] R. Fuchs and H. Hauser, "Visualization of multi-variate scientific data," *Comput. Graph. Forum*, vol. 28, no. 6, pp. 1670–1690, 2009.

[21] J. M. Favre, "Towards efficient visualization support for single-block and multi-block datasets," in *Proc. IEEE Visualization*, 1997, pp. 425–428.

[22] H. Childs, "An analysis framework addressing the scale and legibility of large scientific data sets," Ph.D. dissertation, Computer Science Department, University of California, Davis, 2006.

[23] L. A. Treinish, "A function-based data model for visualization," in *Proc. IEEE Visualization*, 1998, pp. 73–76.

[24] R. Cooke and J. van Noortwijk, "Graphical Methods for Uncertainty and Sensitivity Analysis," in *Sensitivity Analysis*, Saltelli, Chan, and Scott, Eds. Wiley, 2000, pp. 245–266.

[25] T. Nocke, M. Flechsig, and U. Böhm, "Visual exploration and evaluation of climate-related simulation data," in *Proc. Winter Simulation Conference*, 2007, pp. 703–711.

[26] K. Potter *et al.*, "Ensemble-Vis: a framework for the statistical visualization of ensemble data," in *Proc. IEEE Intl. Conf. on Data Mining Workshops*, 2009, pp. 233–240.

[27] J. Kehrer, P. Filzmoser, and H. Hauser, "Brushing moments in interactive visual analysis," *Comput. Graph. Forum*, vol. 29, no. 3, pp. 813–822, 2010.

[28] D. Kao, A. Luo, J. Dungan, and A. Pang, "Visualizing spatially varying distribution data," in *Proc. Intl. Conf. Information Visualization (IV '02)*, 2002, pp. 219–225.

[29] D. Kao, J. L. Dungan, and A. Pang, "Visualizing 2D probability distributions from EOS satellite image-derived data sets: a case study," in *Proc. IEEE Visualization*, 2001, pp. 457–460.

[30] 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.

[31] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, pp. 12–16, 1978.

[32] A. E. Lie, J. Kehrer, and H. Hauser, "Critical design and realization aspects of glyph-based 3D data visualization," in *Proc. Spring Conf. on Comput. Graph. (SCCG 2009)*, 2009, pp. 27–34.

[33] J. Jeong and F. Hussain, "On the identification of a vortex," *J. Fluid Mechanics Digital Archive*, vol. 285, pp. 69–94, 1995.

[34] A. Unger, P. Muigg, H. Doleisch, and H. Schumann, "Visualizing statistical properties of smoothly brushed data subsets," in *Proc. Intl. Conf. Information Visualization (IV '08)*, 2008, pp. 233–239.

[35] N. Boukhelifa and P. J. Rodgers, "A model and software system for coordinated and multiple views in exploratory visualization," *Information Visualization*, vol. 2, no. 4, pp. 258–269, 2003.

[36] E. Chi, "A taxonomy of visualization techniques using the data state reference model," in *Proc. IEEE InfoVis*, 2000, pp. 69–75.

[37] D. Bauer and R. Peikert, "Vortex tracking in scale-space," in *Proc. Symp. on Visualization (VisSym)*, 2002, pp. 233–240.

[38] D. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in visual data analysis," in *Proc. Intl. Conf. Information Visualization (IV '06)*, 2006, pp. 9–16.

[39] E. Bauer, A. Ganopolski, and M. Montoya, "Simulation of the cold climate event 8200 years ago by meltwater outburst from Lake Agassiz," *Paleoceanography*, vol. 19, 2004.

[40] R. Maronna, D. Martin, and V. Yohai, *Robust Statistics: Theory and Methods*. John Wiley & Sons, 2006.

[41] C. Brewer, "Color use guidelines for data representation," in *Proc. of the Section on Statistical Graphics*, 1999, pp. 55–60.

[42] J. W. Tukey, *Exploratory Data Analysis*. Addison-Wesley, 1977.

Johannes Kehrer received a M.Sc. in computer graphics and digital image processing from the Vienna University of Technology, Austria. He is currently a PhD candidate at the Department of Informatics, University of Bergen, Norway. His research interests include the interactive visual analysis and exploration of higher-dimensional, multi-variate, and time- dependent data (primarily from the application fields of climate research and meteorology). He has cooperated with several climate and meteorological institutes.

Philipp Muigg received a M.Sc. in computer graphics and digital image processing from the Vienna University of Technology, Austria. He is currently employed by the SimVis GmbH where he is working on the SimVis framwork. Furthermore, he is a PhD candidate at the Institute for Computer Graphics and Algorithms, Vienna University of Technology, Austria. His research interests mainly lie in the area of interactive visual analysis for scientific data. This includes the development of scalable InfoVis methods as well as GPU based (unstructured grid) volume rendering approaches.

Helmut Doleisch studied computer science at Vienna University of Technology, Austria, and graduated in 1999. After working for two years as university assistant at the Institute of Computer Graphics at the same university, he joined the VRVis Research Center, Vienna, Austria, in late 2000. There he researched the basic SimVis-Technology while working on his PhD. After receiving his PhD in 2004, he headed the "Visual Interactive Analysis" group at the VRVis Research Center as a Key Researcher, supervising students and leading multiple application-oriented research projects in the scope of interactive visual analysis of large and complex simulation data. In 2008, Helmut Doleisch founded the spin-off company SimVis GmbH (www.SimVis.at) where he holds the position of the CEO since then. His research interests are in the areas of interactive visual analysis, flow visualization, and information visualization, and more specifically, in the visualization of large amounts of simulation results.

Helwig Hauser graduated from Vienna University of Technology, Austria, in 1995 and received the degree of Dr.techn. (PhD) from the same university in 1998. He worked as an assistant professor at the Institute of Computer Graphics at the same university from 1994 until mid-2000. Afterwards, Helwig Hauser joined the newly founded VRVis Research Center in Vienna, Austria, as a key researcher in the field of visualization. In 2003, he became the scientific director of VRVis (www.VRVis.at) for four years. Since 2007, Helwig Hauser is professor at the University of Bergen, Norway, within a newly developed research group on visualization (www.ii.UiB.no/vis). His interests are diverse in visualization and computer graphics, including interactive visual analysis, illustrative visualization, the application of visualization to various domain problems, and the combination of scientific and information visualization.