Interactive Focus+Context Analysis of Large, Time-Dependent Flow Simulation Data

Helmut Doleisch, Helwig Hauser, Martin Gasser, Robert Kosara

***Abstract*— Visualization of time-dependent simulation data, such as datasets from CFD simulation, still is a very challenging task. In this paper, we present a new approach to the interactive visual analysis of flow simulation data which is especially targeted at the analysis of time-dependent data. It supports the flexible specification and visualization of flow features in an interactive setup of multiple linked views. Special emphasis is put on new mechanisms to capture *time-dependent features*, i.e., flow features which are inherently dependent on time. We propose the integration of *attribute derivation* into the process of interactive visual analysis to enable the subsequent user access to otherwise implicit properties of the unsteady data in our interactive feature specification framework. All views of this flow analysis setup are linked in the sense that the features in focus are consistently emphasized in the visualization (more colorful, less transparent) whereas the rest of the data is only shown as context in reduced style. In addition to introducing our new approach, we also demonstrate its use in the context of several application examples.**

***Index Terms*— interactive feature specification, time-dependent features, focus+context visualization, time-dependent flow visualization, multi-dimensional data visualization.**

I. Introduction

THESE days, many researchers and practitioners again focus on the visual analysis of results from computational flow simulation, since it eventually has become possible to investigate realistic flow scenarios in three spatial dimensions, which vary over time, and which are represented as large and multi-variate datasets, comprised of dozens of attribute dimensions per location in space and time. It is a special perceptual challenge to effectively convey multi-dimensional and multi-variate data with visualization. Feature-based approaches [26] are attractive because they allow to focus on specific subsets of the flow (*features*), while setting aside the rest of the data, and to thereby enable a targeted and detailed analysis of the flow data. In figure 1, for example, the effects of a flood (caused by a breaching dyke) are analyzed in the vicinity of a nearby object by visually focussing onto the temporal changes of the water level as the time-dependent feature in this case.

In the following, we discuss different approaches to feature-based flow visualization and address related work, before we propose our new approach and discuss our interactive feature specification framework.

All authors have been affiliated with the VRVis Research Center in Vienna, Austria, Donau-City-Strasse 1, A-1220 Wien, `www.VRVis.at`, during this work. Helwig Hauser's new affiliation is the University of Bergen, Norway, `www.uib.no`; Martin Gasser's new affiliation is the Austrian Research Institute for Artificial Intelligence in Vienna, Austria, `www.ofai.at`; Robert Kosara's new affiliation is the University of North Carolina at Charlotte, NC, USA, `www.uncc.edu`; the authors can be contacted via {Doleisch,Gasser,Kosara,Hauser}@VRVis.at

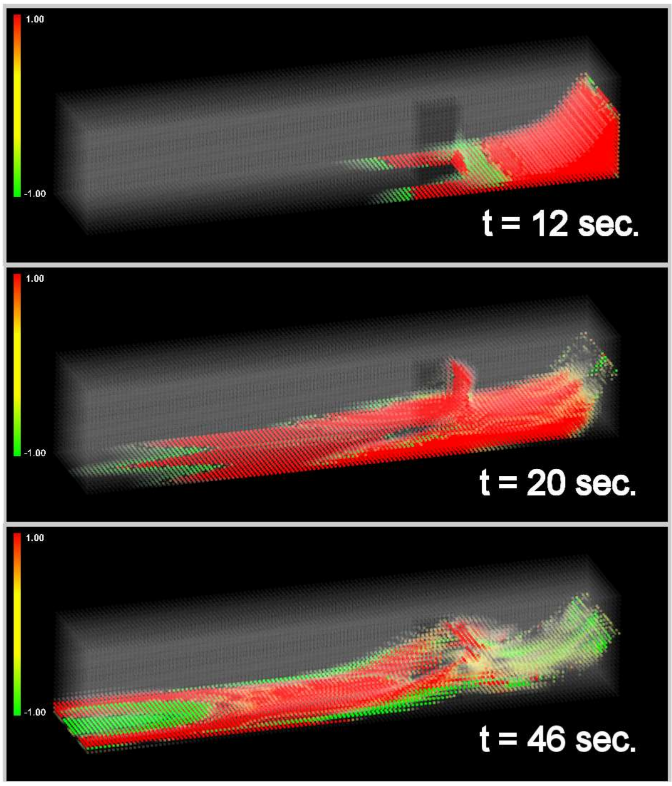

Fig. 1. The effect of a flood, resulting from the burst of a dam (on the right), is visualized in a feature-based 3D visualization. Grid locations which exhibit a temporal maximum (red) or minimum (green) wrt. the percentage of water in the cell have been specified as the flow feature in focus.

A. Approaches to feature-based flow visualization

Gaining thorough insight into flow simulation data is indeed very challenging. One reason for this is that not all of the interesting information in the data also is explicitly represented in the form of first-order data attributes in the files. Flow vortices, for example, usually are not directly accessible through explicit data attributes, but implicitly represented in the data. Taking this consideration a step further, we can think of the information content of such a flow simulation dataset as being given in a *layered feature space*, where deep layers stand for complex (and implicit) relations in the data, e.g., a vortex or a shock wave, and where shallow layers correspond to data properties which are explicitly represented in the form of first-order data attributes (such as temperature values). Figure 2 illustrates this layered feature space with the first-order data attributes forming the upper most layer and the more complex flow features being considered as the "deep treasures" in the

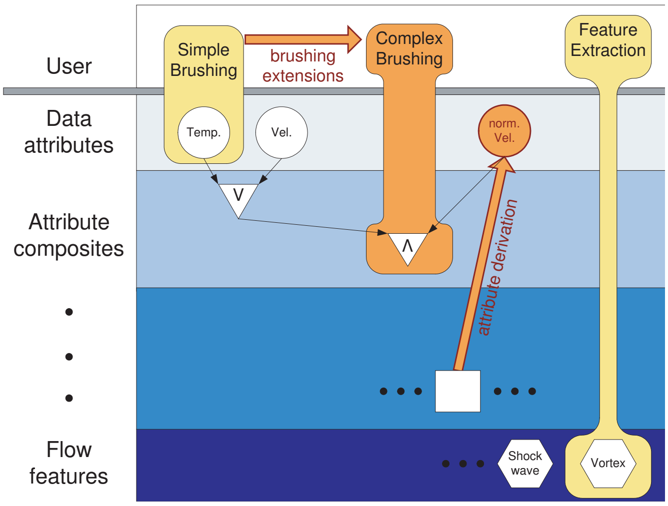

Fig. 2. The layered feature space: attribute derivation and extended brushing form a hybrid, feature-based visualization approach.

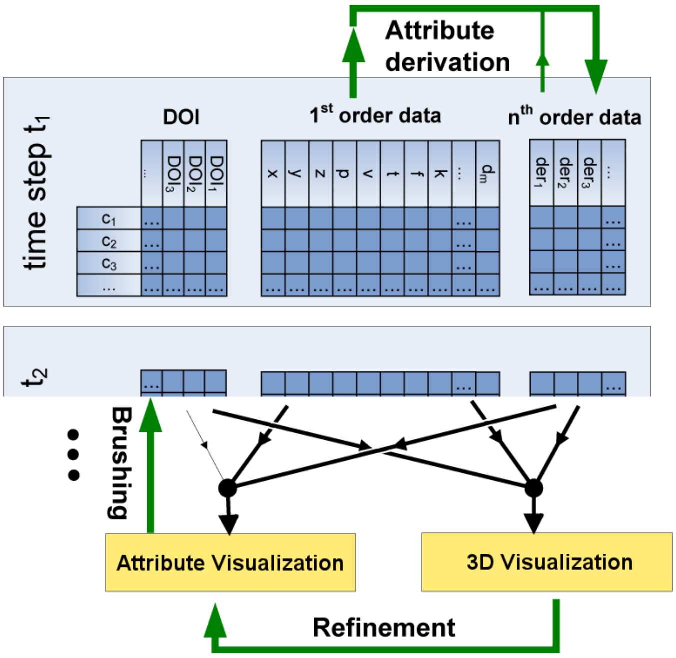

Fig. 3. Scheme for the interactive specification and visualization of time-dependent flow features: interplay of *attribute derivation*, *interactive brushing*, and *iterative refinement* in the context of the multi-variate flow data representation.

layered feature space.

Feature-based flow visualization requires that the user has access to data properties, either in the form of first-order data attributes or from deeper layers in the layered feature space. A very effective solution for this feature access is to support interactive brushing [32] of data attributes – brushing means that the user can interactively mark up the interesting parts of the data by interactively "brushing" their visual representation in the visualization view. Given a scatterplot, for example, with temperature values relating to the y-coordinates of all points, a user could interactively select the hot subsets of the data as the current feature of interest by just drawing a rectangular brush over the upper parts of this scatterplot. In according visualization systems, a set of advanced interaction techniques is available to ease the interactive brushing of first-order data attributes (see related work section).

Additionally, there also are advanced approaches to extract the "deep treasures" from the layered feature space, i.e., to access and visualize also complex relations within the data. Usually, sophisticated computations are used to extract those parts of the data which correspond to the respective flow features. Special challenges include the extraction of non-local relations, computationally expensive extraction methods, and numerically unstable feature specifications. Often, feature extraction techniques also pose non-negligible challenges for the user to fully comprehend what the feature extraction process actually is doing (and which result it actually delivers). In many cases, the extracted flow features are explicitly represented in the visualization, for example, by the means of boundary surfaces or glyphs. Often, the rest of the data is not shown at all, yielding significant compression rates of several orders of magnitude [26], but additionally stressing the challenges of comprehension.

In this paper, we propose a new and hybrid approach to the feature-based visualization of simulation data which allows to utilize the advantages of interactive brushing as well as those of feature extraction.

On the one hand, we "lift up" implicit data properties from the deep levels in the layered feature space by *attribute derivation*. For every data item c_i, we compute (on demand) additional, synthetic data attributes (2^{nd}- or n^{th}-order data attributes) as mathematical combinations of other, already given or previously computed data attributes as illustrated in figure 3. This happens, for example, through the application of specific filters or by the use of formulas which describe the physical/chemical relations of interest within the flow data.

On the other hand, we provide the user with *advanced brushing mechanisms* so that the flow features can be easily grasped, even from lower levels in the layered feature space, and taken further to expressive focus+context visualization. Logical combinations of selections (brushes), which are formulated with respect to 1^{st}-, 2^{nd}-, or n^{th}-order data attributes, can be used to access also fairly complex relations (features) as compared to traditional, simple brushing.

One advantage of our approach is that users can easily understand how the feature-based visualization is constructed and how the results are to be interpreted. To analyze the effectiveness of a cooling jacket, for example, a user would first use attribute derivation to represent local temperature maxima in the form of a new synthetic data attribute, and then select all those data subsets for subsequent visual analysis which exhibit both high temperature maxima and low flow velocities in the same places (locations of insufficient cooling).

This approach provides a large amount of flexibility during feature specification, supports the comprehension of the feature extraction process, and is well-aligned with focus+context visualization, where features are visually emphasized as compared to the rest of the data, which only is provided as context in reduced style for improved orientation and navigation.

B. Related work

In the field of flow visualization, many different approaches have been proposed up to now, especially for 2D flows and for steady flows. A detailed overview about different visualization methods and feature extraction techniques is available in an overview paper by Post et al. [25].

For the visualization of *3D unsteady flows*, also several approaches have been presented. One way to visualize unsteady flow data is the use of (moving) isosurfaces or arrows on boundary surfaces as presented by Treinish in a visualization of unsteady weather data [31]. Another approach is to use volume visualization. Swan et al. apply direct volume rendering techniques to flow visualization in a system which supports computational steering [30]. Clyde and Dennis [5] as well as Lum et al. [21] present volume rendering for time-varying vector fields using algorithms which make special use of graphics hardware to accelerate the rendering process.

In contrast to direct volume rendering, feature-based approaches profit from an early reduction of the data (ahead of visualization) so that not all of the data is visualized concurrently in full detail. For feature-based visualization, proper feature extraction is essential. Depending on the actual application, flow features typically include vortices, shock waves, separation and attachment structures, recirculation zones, etc. [26]. Many different approaches to feature extraction have been proposed. For the detection of regions of high vorticity or vortex core lines, Kenwright and Haimes [19], Bauer et al. [4], and Roth and Peikert [23][28] proposed advanced approaches. Additionally, many other feature extraction methods have been presented – due to space limitations we provide a more thorough review in another paper [26].

In our case, an additional challenge is to include the temporal dimension into the feature specification process. Most feature extraction techniques for unsteady data extract features in individual time steps [27][29]. Usually, feature tracking is then used to match the features of successive time steps and thereby reconstruct a notion of time-dependent features.

Up to now, only little work has been done on interactive feature specification. Henze developed a technique to visually analyze time-varying data from CFD simulation (computational fluid dynamics) by using multiple, linked portraits (2D plots, also showing the grid connectivity of the data) in a system called Linked Derived Spaces [17]. The system supports a basic brushing and linking functionality and is designed for exploratory visualization and feature detection in multivariate data. Gresh et al. presented a system called WEAVE [14], which combines the visualization of simulation data with a 3D visualization of the simulated structure and which allows for structural investigation of complex simulation data in the 3D context, based on marking data items according to different data attributes. Inspired by this work, we developed a framework for interactive feature specification for CFD data in previous work [7][6], called SimVis [2] (see also section II).

In the following, we first briefly describe our feature-based flow analysis framework in general before we detail on our new approach to time-dependent flow features. Additionally, we also describe some recent extensions to the focus+context visualization as well as performance improvements, which have become necessary to handle large datasets. Finally, we demonstrate our approach in the context of two applications.

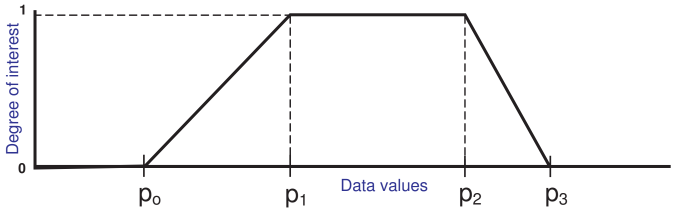

Fig. 4. A 1D smooth brush results in a trapezoidal DOI function according to equation 1.

II. The SimVis System

To achieve feature-based flow visualization for data exploration and analysis, we provide 3D flow visualization together with different kinds of attribute views. Advanced interaction mechanisms enable the user to intuitively specify features in the flow data. Multiple, linked views (compare to Baldonado [3]) are used to concurrently show different aspects of the flow data. In the visualization, the flow features are visually discriminated from the rest of the data in a focus+context visualization style which is consistent in all views [15].

For interactive feature specification, attribute visualization views such as scatterplots [24], histograms [20], etc., are used. The user chooses to visually represent selected data attributes in such a view, thereby gaining insight into the selected relations within the data. Then the interesting subsets of the data are interactively brushed directly on the screen (compare to the XmdvTool [32], and see figures 5c and 5d). The result of such a brushing operation, i.e., a notion of user interest, is reintegrated within the data in the form of a synthetic data attribute $DOI_j \in [0,1]$ (*degree of interest (DOI) attribution* of the data, compare to Furnas [11]).

The SimVis system supports *smooth brushing* [7] (to enable fractional DOI-values) as well as the logical combination of brushes for the specification of *complex features* [6]. A smooth brush results in a trapezoidal DOI function as defined below and illustrated in figure 4.

$$DOI_j(d_i, p_0, p_1, p_2, p_3) = \begin{cases} 0 & \text{if } d_i \leq p_0 \\ \frac{d_i - p_0}{p_1 - p_0} & \text{if } p_0 < d_i \leq p_1 \\ 1 & \text{if } p_1 < d_i \leq p_2 \\ \frac{p_3 - d_i}{p_3 - p_2} & \text{if } p_2 < d_i \leq p_3 \\ 0 & \text{if } d_i > p_3 \end{cases} \quad (1)$$

In SimVis, DOI attributes and their compositions are explicitly represented in the system [6] and can be interactively adjusted through a separate graphical user interface to achieve comparisons – DOI attributes can be saved and reapplied to other datasets (see figure 5a).

For feature-based flow visualization, SimVis provides 3D focus+context visualization of the flow data in the sense that flow features are visually emphasized in the 3D depiction through coloring and reduced transparency [15]. For each 3D view, one (fractional) DOI attribute DOI_j is used to

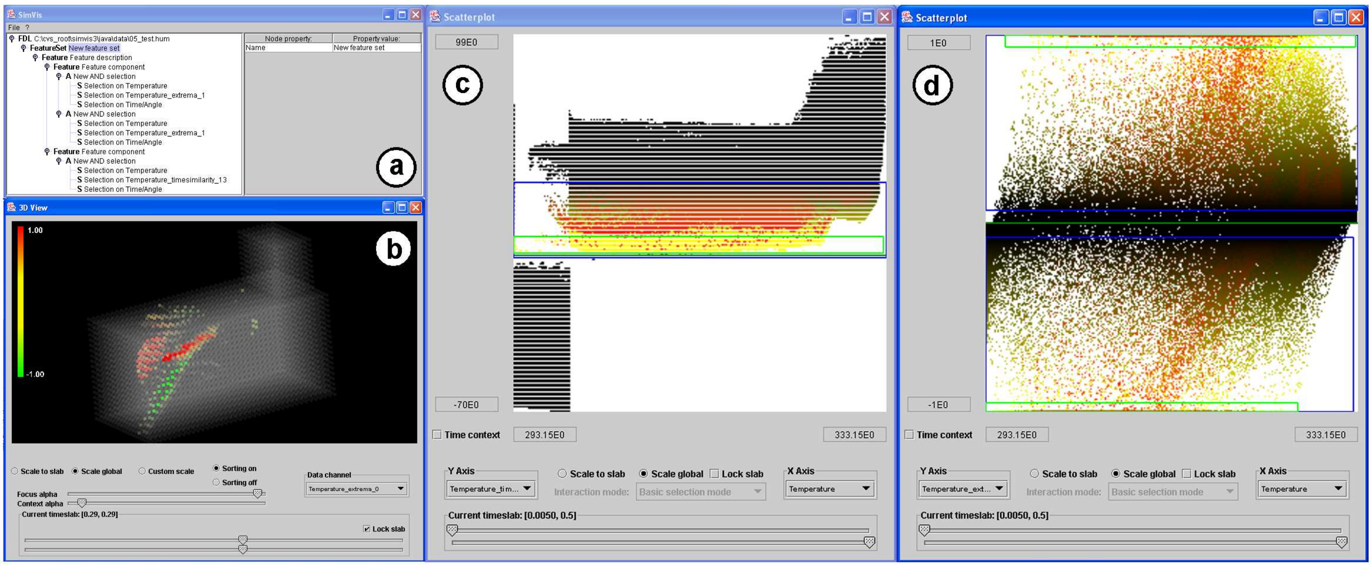

Fig. 5. Typical setup of a working session with the interactive feature specification framework: a tree viewer is used to manage the feature specification (a); a 3D visualization view, used to visualize the spatial structure of data using a focus+context visualization approach (b); two scatterplots, showing interactively brushed and refined feature specification parts (c&d).

(smoothly) discriminate the visually enhanced focus from the rather transparent and grey-scale context (see figure 5b).

From the collaboration with our industrial partners we know that interactive exploration and analysis of simulation data often follows one of three characteristic interaction patterns. One type of exploration is to search for places in the 3D simulation grid where certain feature characteristics are present (*feature localization*). In the SimVis system, the user can brush features in attribute views and concurrently localize the respective feature in the 4D (3D+time) flow domain.

Additionally, users also investigate multi-variate data properties by specifying a feature in an attribute view and at the same time analyzing the DOI distribution with respect to other data attributes in other attribute views (through view linking, here called *multi-variate analysis*). This sort of analysis also is inherently supported by the SimVis system.

Finally, users also often inspect the values of selected data attributes with respect to certain spatiotemporal subsets of the flow domain (*local investigation*). In the SimVis system, the user can also load spatial as well as temporal data references into attribute views – brushing these kinds of data attributes then yields features which are specified according to their spatiotemporal extents.

III. Interactive Specification of Time-Dependent Features

When analyzing datasets from unsteady CFD simulations, interactive visualization is very useful. Flexibility with respect to feature specification also is of great utility for the user. For time-dependent data it is necessary to support the specification of features which are inherently based on time. In the following, we call features which cannot be extracted from singular time steps of unsteady CFD data, *time-dependent features*.

As a result from an informal user study with application engineers we identified five types of important time-dependent features. In the following subsections we discuss these feature types one by one. In the SimVis system, attribute derivation is issued through the use of a separate interface – the user chooses which data attribute to derive from, the type of derivation, and adjusts derivation parameters. For the subsequent visual analysis through interactive brushing, all data attributes (1^{st}- as well as n^{th}-order) are equally available.

A. Features based on attribute derivatives

Often, users are interested in changes of data attributes d_i with respect to time. So we provide the user with access to attribute derivatives ($d'_i(c,t) = \mathrm{d}\, d_i(c,t) \,/\, \mathrm{d}\, t$). Due to the discrete representation of time in the form of time steps, differential information is approximated, usually in one of three typical forms: forward, backward, or central differences. In the SimVis system, central differences are used as a useful compromise between data smoothing and frequency amplification (side-effect of data derivation) according to the following formula.

$$\delta\, d_i(c,t_k) = \frac{d_i(c,t_{k+1}) - d_i(c,t_k)}{2(t_{k+1}-t_k)} + \frac{d_i(c,t_k) - d_i(c,t_{k-1})}{2(t_k - t_{k-1})} \quad (2)$$

With this type of derivatives estimation we compensate for cases of unevenly spaced time steps in the simulation.

Derivatives are used to detect patterns of changes. One example is a search for large temperature changes when investigating the burning front in a combustion chamber.

Iterative attribute derivation also is an important functionality of our approach. Applying derivatives estimation repeatedly, for example, results in second-order (or higher-order) differences. Figure 6 gives an example of how attribute derivatives of different orders are used for feature specification. In a 3*2D+1*3D scatterplots view, the simulation time (red axis), normalized first-order derivatives of temperatures (green axis), and second-order derivatives of temperatures (blue axis) are plotted against each other for the flow dataset from figure 5.

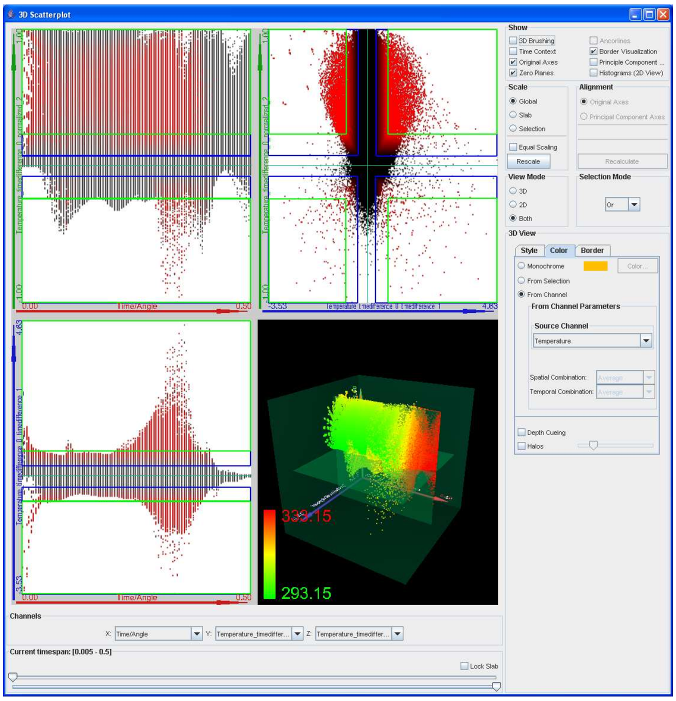

Fig. 6. Specifying a feature in a 3*2D+1*3D scatterplots view: normalized 1st-order derivatives (green axis) and 2nd-order derivatives (blue) of temperature values are plotted over time. Higher values of both derivatives are brushed in the upper right scatterplot. For a related 3D visualization see figure 7.

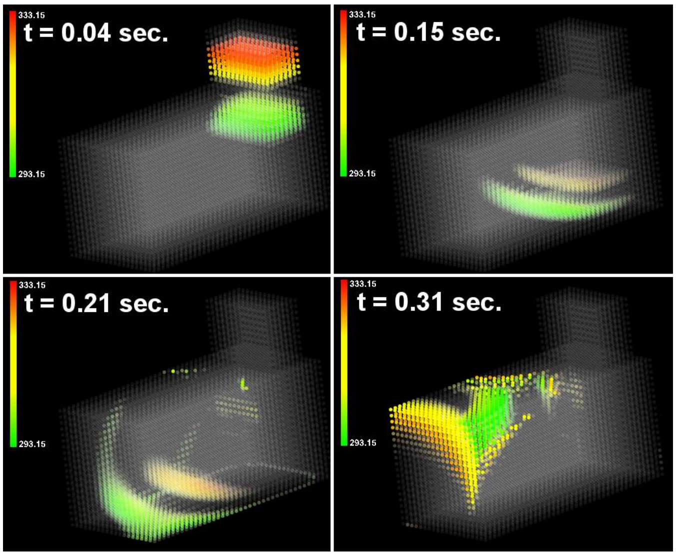

Fig. 7. Areas in front and behind of a hot inflow front in the dataset of fig. 5, visualized over time. The related feature specification is shown in fig. 6.

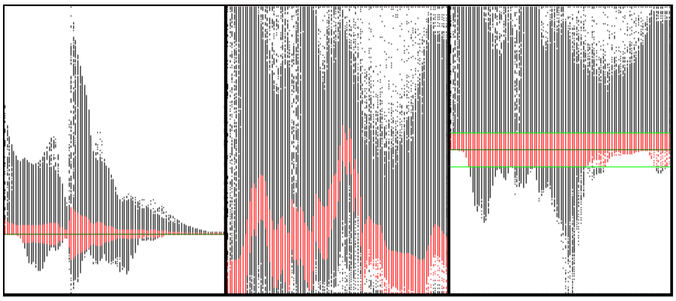

Fig. 8. Comparison of derivatives (left), simple normalized derivatives (middle) and zero-preserving normalized derivatives (right). The data shown are temperature derivatives in the extended T-junction example from section VI.

A complex brush is used to investigate flow regions in front and behind a moving hot inflow front (see figure 7).

Another case of iterative attribute derivation is to first filter the data (with respect to time) in a step prior to attribute derivation. As one often demanded operation, we provide smoothing of the data according to

$$\Gamma d_i(c,t) = \frac{\sum_{t-w \le \tau_j \le t+w} d_i(c,\tau_j) \cdot G((t-\tau_j) \cdot 4/w)}{\sum_{t-w \le \tau_j \le t+w} G((t-\tau_j) \cdot 4/w)} \quad (3)$$

with t denoting the time of the current data item of interest. As a filter a Gaussian lobe with σ=1 is used from $[-4, 4]$, scaled to time-interval $[-w, w]$. For other purposes, of course, $G()$ also can be exchanged with any other filter kernel as needed.

B. Relative feature specifications

Usually, the per-time-step min–max range of attribute values varies over time – the meaning of what is relatively hot (at an instance of time) changes as the distribution of temperature values varies over time. Accordingly, we enable user access to relative attribute measures d_rel_i, which are computed with respect to the per-time-step data range of the data attribute d_i at the time step t, and thereby specify time-step-relative features:

$$d\_\mathrm{rel}_i(c,t) = \frac{d_i(c,t) \; - \; \min_c d_i(c,t)}{\max_c d_i(c,t) \; - \; \min_c d_i(c,t)} \quad (4)$$

An example is the investigation of relatively high temperature changes by the use of derivatives estimation first and a subsequent selection of relative extrema.

For the specification of features relative to data changes, local data normalization with respect to time is necessary. In SimVis, data normalization can be applied to any data attribute available during investigation. Two conceptually different versions are available: the local data range of each time step can be simply mapped to the unit-interval as shown in equation 4. Alternatively, normalization can be done to the $[-1, 1]$ interval such that zeros in the input data also map to zeros in the output as described by the following four cases ($p \ge 0, k \ge 0$):

$$\begin{array}{ll} 1: & [+p, +k] \text{ with } 0 \le p < k \text{ is mapped to } [+p/k, +1] \\ 2: & [-p, +k] \text{ with } p \le k > 0 \text{ is mapped to } [-p/k, +1] \\ 3: & [-p, +k] \text{ with } p > k \ge 0 \text{ is mapped to } [-1, +k/p] \\ 4: & [-p, -k] \text{ with } p > k > 0 \text{ is mapped to } [-1, -k/p] \end{array}$$

Figure 8 illustrates the two different versions of normalization and how they compare to each other. In the most left scatterplot, temperature derivatives are shown on the vertical axis vs. time steps on the horizontal axis. Simple normalization of the derivatives is presented in the middle whereas zero-preserving normalization is shown in the right scatterplot.

With the help of this type of attribute derivation, *relative brushing* of certain data attributes can be realized very easily. After data normalization of a specific data attribute, a brush

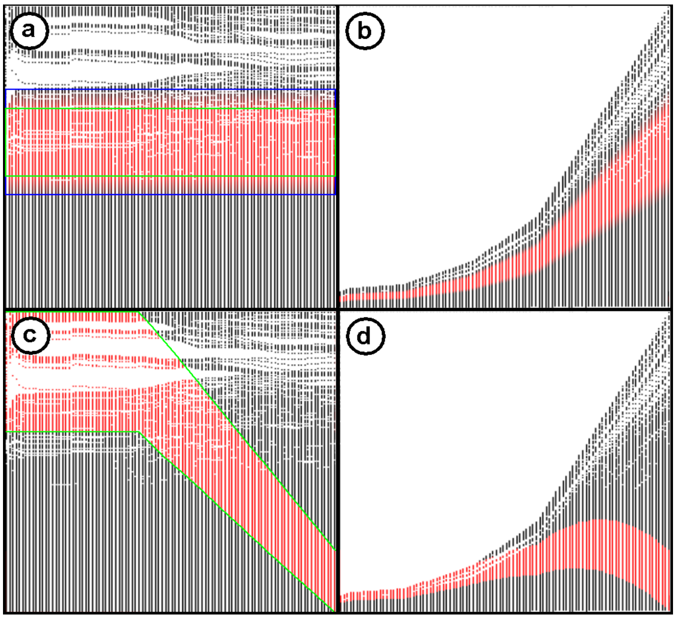

Fig. 9. Comparing feature specification relative to data changes (relative brushing, a&b) with interest which varies over time (interpolating brushes, c&d). The different scatterplots show: normalized velocity values for each time step (a&c) and original velocity values over time (b&d). The time domain is always on the x-axis, the data is the same as in figure 5.

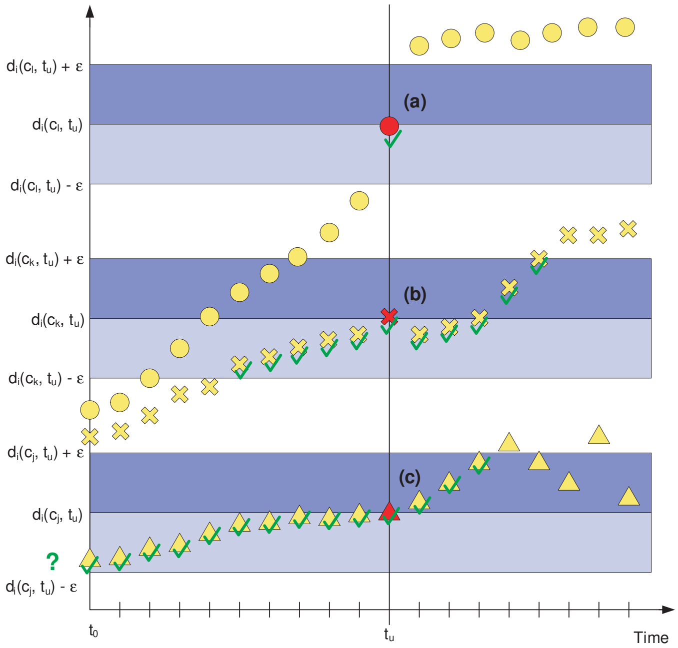

Fig. 10. Illustrating different cases for the calculation of stationary attribute values. In (a) the attribute is not stationary, in (b) the attribute stays stationary for 5+1+5=11 time steps, and in (c) the attribute stays stationary for 10+1+3 time steps (or possibly even longer while outside the simulation domain).

can be specified in the normalized data range, always selecting the same relative data range per time step. Figure 8 shows how derivatives relatively close to zero have been brushed in the right scatterplot, the other views are visually linked.

C. Interest which varies over time

Due to simulation processes which cover certain temporal phases of the flow, users sometimes are interested in features whose specifications vary from phase to phase. The difference to feature specification relative to data changes is that also the definition of the (relative) interest changes over time and not only the data range of the attribute. Providing an interpolation scheme for feature specifications, a continuous change of feature specifications between selected key time-steps is realized. Interpolation between the DOI specifications for two different key time-steps s and t is enabled by extending the DOI definition from equation 1 to

$$DOI\_\mathrm{var}_k(d_i, s, \mathbf{p}_s, u, \mathbf{p}_u) = DOI_j(d_i, \mathbf{p}_s + \frac{t-s}{u-s} \cdot (\mathbf{p}_u - \mathbf{p}_s)) \tag{5}$$

where $DOI_j(d_i, \mathbf{p}) = DOI_j(d_i, p_0, p_1, p_2, p_3)$ and $s \leq t \leq u$.

An example of an interest which varies over time is the definition of moving brushes in the spatial domain when working with datasets which are based on time-varying grids. In this case, parts of the geometry are interpolated from key-geometry to key-geometry. With an interpolation scheme for brushes, spatially moving parts of the geometry can be kept in focus for further feature refinement in the respective areas.

To illustrate the difference between feature specification relative to data ranges and an interest which varies over time, a comparison is presented in figure 9. Data is shown from the example of figure 5. In all four scatterplots time is shown on the x-axis, (a) and (c) show normalized velocity values on the y-axis, (b) and (d) original velocity values. In figure 9(a) a relative brush is specified, selecting relatively medium-high velocity values. The resulting DOI distribution is visualized in (b) according to the original velocity values. In (c), a time-varying interest has been specified, brushing relative high values of velocities for the first temporal phase of the dataset, and then decreasingly lower relative values for the rest of the time. The resulting DOI-definition with respect to the original velocity values is shown in figure 9(d).

D. Features based on stationary attributes

In contrast to the interest in changing phenomena, users are sometimes also interested in subsets of the flow, where certain attributes remain stationary (for a certain time and within a certain tolerance). We attribute each data item (at each instance of time) with the number of time steps $\mathrm{stat\_count}_j$ (equ. 6), for which a certain data attribute d_i remains stationary with respect to a user-defined threshold ε. Features with respect to such stationary attributes are then easily defined.

$$\mathrm{stat\_count}_j(d_i(c,t), \varepsilon) = \max_{s \leq t \leq u} \| \{ s, \ldots, u \} \| \mid$$
$$\forall \tau \in \{s, \ldots, u\} \colon d_i(c,t) - \varepsilon \leq d_i(c,\tau) \leq d_i(c,t) + \varepsilon \tag{6}$$

Figure 10 illustrates three different examples of how the calculation of *measures for stationary flow attributes* ($\mathrm{stat\_count}_j$) is performed. In (a), the data value of cell c_l lies in the specified ε-neighborhood only for time step t_u. In this case, $\mathrm{stat\_count}_j(d_i(c_l, t_u), \varepsilon)$ is assigned the value of 1, which means that the data value is not stationary at all (but stays only for one time step in the specified data range). Case (b) illustrates an example, where the data values of cell c_k stay for a longer time within the specified data range (here for 5 preceding and 5 succeeding time steps). Accordingly, $\mathrm{stat\_count}_j(d_i(c_k, t_u), \varepsilon)$ is assigned to the value of 5+1+5=11. A special case of our calculation scheme is shown in case (c). If the temporal boundary of the simulation is reached (the first or the last time step) before attribute values leave the ε-neighborhood of d_i, then we tag $\mathrm{stat\_count}_j(d_i(c_j, t_u), \varepsilon)$ by representing it as a negative number. This way, we are able to separately treat all those cases where the stationary measure potentially is even larger than actually computed. Here in case (c), $\mathrm{stat\_count}_j(d_i(c_j, t_u), \varepsilon)$ is assigned the value of $(10+1+3) \cdot -1 = -14$, accordingly.

An example for interest in features which are based on stationary attributes is the investigation of all those regions of a cooling jacket for a Diesel engine where a stationary high temperature of the cooling fluid is given.

E. Features based on local extrema

Often, it is of special interest where and when local extrema occur in a dataset, e.g., when temperature reaches a temporal maximum in a specific location of the simulation grid. In such cases, not only the maximal/minimal value of the respective data attribute is of interest, but also the spatial location as well as the point in time when the local extremum happens. Below, we describe how DOI values can be computed to represent local data extrema for each cell c.

$$\begin{aligned} d\_\mathrm{lmax}_i(c,t) = \min(&DOI_j(\delta\Gamma d_i(c,t), -\varepsilon_1, -\varepsilon_0, \varepsilon_0, \varepsilon_1), \\ &DOI_j(\delta\delta\Gamma d_i(c,t), -\infty, -\infty, \gamma_0, \gamma_1)) \end{aligned} \tag{7}$$

In this equation $\Gamma d_i(c,t)$ represents smoothed data values as defined by equation 3, δ denotes differentiation wrt. time, and $DOI_j()$ represents a subset selection. The boundary representations ε_0 and ε_1 as well as γ_0 and γ_1 influence how sharply local maxima are specified with $\gamma_0 < \gamma_1 \leq 0$ and $\varepsilon_1 > \varepsilon_0 \geq 0$. The minimum operator is the traditional T-norm which represents a logical AND-combination in fuzzy logic. Equation 7 represents nothing else than the fact, that local maxima are found where 1^{st}-order derivatives are zero (smoothly brushed) and 2^{nd}-order derivatives are negative (also smoothly brushed).

For calculating local minima, only the second argument of the $\min()$-operator in equation 7 has to be replaced by $DOI_j(\delta\delta\Gamma d_i(c,t), \gamma_1, \gamma_0, \infty, \infty)$ with $\gamma_0 > \gamma_1 \geq 0$.

One example of interesting local maxima is again the cooling jacket for a Diesel engine. Here it would be easy to find all those moments in time, when temperature values of the cooling fluid are at a maximum (and above some limit).

IV. Visualization Challenges

All the above described feature types can be interactively specified in the attribute views of our system. If attribute derivation is performed during analysis, the resulting derived data attributes can be used in any of the linked views of our framework just as first-order data. In the following, we describe how SimVis views were extended to cope with time-dependent flow data.

Temporal focus+context visualization

When dealing with unsteady datasets, usually all data attributes are provided for multiple time steps (usually for all of them, if no pre-selection is performed before the analysis). When viewing the data, two straight-forward options are to either show the data of all time steps simultaneously, or to interactively select one time step for visualization.

In the SimVis system, the user can choose in each view from which specific time steps data items are visualized by adjusting two sliders. Usually both sliders are bound to each other, so that data from just one time step is shown. Alternatively, a from–until span of time steps can be selected, enabling the concurrent visualization of data from multiple (or all) time steps. An example is shown in figure 5. In the 3D visualization on the left side, only the data from time step 14 is selected for rendering, whereas in the two scatterplots on the right side, data from all time steps is shown.

For better control of the time step selection we propose a commonly used VCR-like interface. It enables the synchronized animation of all views in the system through simultaneously updating the temporal focus. It allows to step through the time steps interactively in both directions of time, to animate this process at a fixed step rate, and to rewind and loop animation sequences. For the generation of short repeated animation sequences of especially interesting time spans, all options for animations can be restricted to a specific time interval.

In addition to interactively cycling through time steps, we also provide the integration of data from time steps out of focus as temporal context in attribute visualization views. We follow the focus+context visualization concept for this purpose. All data items in focus from the selected time steps are shown in a prominent style (black points in scatterplots), whereas the data items from the temporal context (not selected time steps) is visually deemphasized (in grey-level style) to just provide an impression of the overall data distribution. This requires, however, that the user pays attention to the fact that during brushing in an attribute view, the brush action only is applied to the data items in focus, but not to the context. Figure 11 shows an example, where only data from a small time interval is in focus.

In SimVis, all views can also display time values on all axes, similar to regular attribute values. Examples include the mapping of time onto the axes of scatterplots (see figure 9) or the color-coding of time when jointly rendering multiple time steps in the 3D view.

Scaling the attribute visualization

In the visualization of time-dependent data, ranges of data attributes usually vary between time steps. When visualizing data from different time steps (or time intervals), this has to be considered. We provide different scaling options in each view: *local scaling* to the data range of the currently visualized time step (or time interval), *global scaling* to the data range of all time steps, *scaling to a specific selection* (scaling to a brush), and *user-defined scaling* (interactive scaling) as illustrated by figure 11. Here velocities (x-axis) vs. temperatures (y-axis)

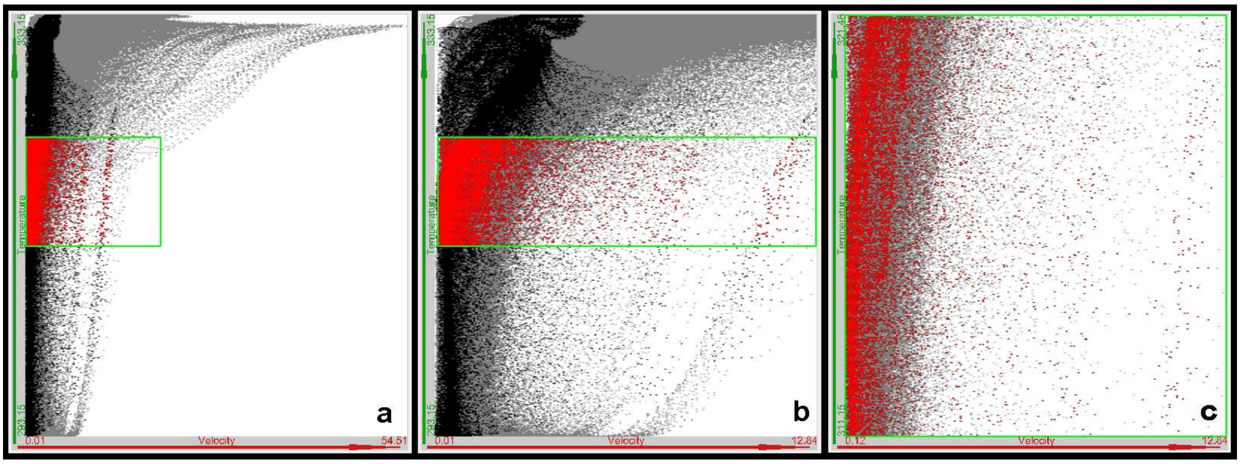

Fig. 11. Illustration of three different types of *view scaling* for the data from figure 5 (velocity vs. temperature in all three scatterplots): (a) scaling to show all data (all time steps); (b) scaling to the data from a selected time interval; (c) scaling to the previously brushed selection (brush from a).

of flow data from figure 5 are plotted in three scatterplots. Figure 11a shows global scaling for the full time range, figure 11b scales to the data ranges of a selected time interval (time steps 48 to 52 out of 100 available time steps), and figure 11c shows the data distribution scaled to the brush, shown in the first two examples.

Multi-level focus+context visualization

Since SimVis utilizes a multi-view approach to focus+context visualization, feature specification is possible in any of the attribute views. SimVis not only provides the opportunity to specify multiple features in multiple views, but especially supports the interactive specification of complex features in two or more views. Degree-of-interest values which stem from (smooth) brushes in one or more views can be AND-, OR-, and SUB-combined, based on fuzzy logic operations [6]. In the focus+context visualization, we differentiate four levels:

- most important (first-level focus) is the DOI attribution which represents the complex feature itself (after logical combinations). In the attribute views, shades of red are used to represent data items in first-level focus.
- within each of the attribute views, all those data items which on the one hand are brushed but on the other hand are not in first-level focus (due to the logical combination with other brushes) are considered as second-level focus. Shades of yellow are used to visually enhance the second-level focus.
- all data items from the time interval, which has been used as the temporal support for the feature specification, and which are not in first- or second-level focus, are considered as first-level context in the visualization and visualized in black.
- all other data items, i.e., the data items outside the selected time interval, are considered as second-level context for visualization and visualized in grey (or left out entirely).

Concerning drawing order in the visualization, we note that first-level focus is drawn over second-level focus, and so on.

Extensions to 3D rendering in SimVis

The rendering of data from single time steps in the 3D rendering view in a step-through or animated mode is straight forward. In contrast, 3D rendering of data from a time interval is a more complicated case. In feature-based rendering, the concurrent visualization of data from multiple time steps allows to show the temporal evolution of features in one single image. Another effect is, that by rendering a few time steps simultaneously, outliers, i.e., parts of features which are present only for a short time, are smoothed out and thereby deemphasized.

When rendering data from a time interval in the 3D visualization, all selected time steps are rendered into one final image. We render the cells sorted (back to front) and for each cell all time steps are rendered and composited. Each time step (out of n) is rendered using a reduced opacity $\beta = 1-\sqrt[n]{1-\alpha}$, so that the overall opacity α (for all time steps) stays the same, whether only one time step is rendered or several. Figure 15 shows a result of such a 3D rendering of several time steps.

V. Large Data Handling

Datasets from CFD simulation usually are large or even huge. The size of the data is influenced by the number of time steps, the number of data attributes at each cell, and the number of grid cells themselves. Dealing with very large datasets during visualization and interaction is a challenging task, therefore optimized data access policies are very important for a useful visual analysis framework like SimVis.

The datasets which we are currently analyzing with the SimVis system have about 100 time steps, 20–50 data attributes and 100,000 to 1,000,000 cells. File sizes of such datasets range from several hundred megabytes to a few gigabytes. The main extensions to the SimVis system for supporting an efficient handling of large datasets are outlined below. Further details are discussed in a separate paper [13].

First, we enabled lazy loading (*activating*) of data attributes. This is realized by implementing a *Virtual Proxy* pattern [12] in the data attribute classes. Due to the fact that it is very expensive in terms of system resources to leave all data

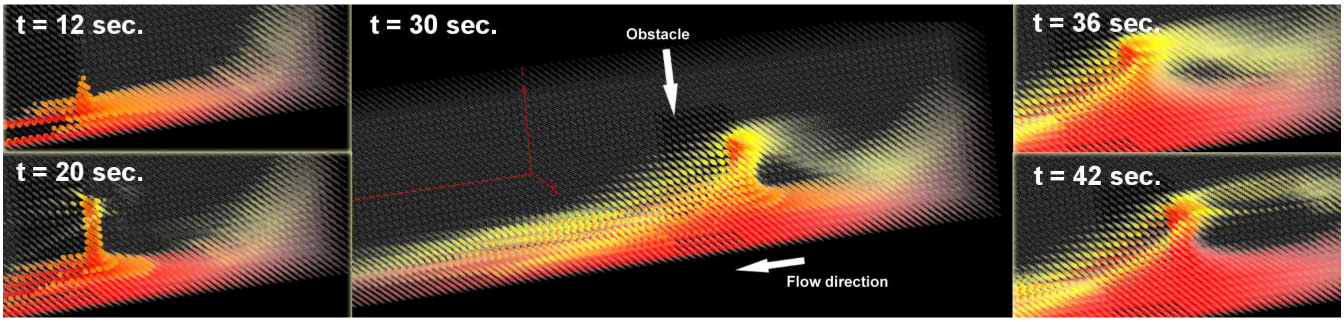

Fig. 12. Using a time-dependent flow feature, based on a stationary flow attribute (large amounts of water), to generate a visualization sequence which well visualizes the impact of a flood (caused by a breaching dyke) onto a nearby object.

attributes permanently activated, a caching algorithm based on a LRU (least recently used) queue has been implemented. First-order data attributes, which can be re-read from the simulation output at any time, are not swapped out to a new external file for later reuse as opposed to data resulting from attribute derivation.

In addition to the improved memory management, SimVis also has been optimized with respect to computational performance. The original version of SimVis was single-threaded, resulting in a blocking behaviour in certain cases – originally about 90% of the computing time was used for the calculation of DOI values and mouse interaction continuously triggered a recomputation of the DOI values. Our solution to maintain an interactive response of the user interface (without blocking), and thus to enable interactive changes of feature specifications, was to decouple the DOI computation from the user interface by using two threads. The DOI computation thread notifies the registered views after finishing the computation to update the viewing information accordingly. DOI computation time is additionally reduced by only recalculating parts of the DOI hierarchy, which are affected by user interactions.

To further speed up computationally intensive processes, we also investigated two hardware-dependent features:

Intel Pentium IV processors support the powerful SSE instruction set (Streaming SIMD Extensions [10]) which can perform calculations on four 32 bit float values simultaneously. It is possible to optimize the calculation of the DOI function based on this extension, since it packs 4 operations into one. On the other hand, the Pentium IV also supports *hyperthreading* technology (HT) [1] which offers thread-level parallelism, enabling multi-threaded applications to make better use of CPU resources as compared to single-threaded applications.

We tested both solutions for their suitability in the special case of calculating combinations of data coming from different large float arrays (as it is the case for the performance-critical parts of the DOI calculation). As a benchmark we used the following case: the minimum T-norm (logical AND) of arbitrary DOI values from two arrays with 50,000 entries each is calculated (similar to the combination of brushes in SimVis [6]). Figure 13 gives averaged results for multiple tests with the different setups (with and without SSE, with and without hyperthreading). Using (only) the SSE instruction set gives the maximum speed-up for our case. When adding

	No SSE	SSE
Without HT	10.36ms	2.81ms
With HT	8.45ms	5.08ms

Fig. 13. Comparing DOI calculation times for different computational setups.

hyperthreading (by using different threads for calculating parts of the combinations, for example), no further speed-up can be achieved. Compared to using only SSE extensions, the increased cache miss numbers due to using different threads even slow down computations significantly in most cases. We therefore optimized the calulation of the DOI function based on the SSE extensions.

The SimVis system currently runs interactively on a standard PC (Intel Pentium4, 2.8GHz, 2GB of memory) with a NVidia GeForceFX 5900 graphics card for datasets of up to 10 million data items (cells*time steps), and 50 to 100 data attributes for each data item.

For the implementation of SimVis, a Java/C++ hybrid approach was chosen. The UI interaction and the handling of the feature specification have been realized in Java, whereas mesh access and the rendering of the visualization views is implemented in native code. Native methods are called via the JNI API. The mesh access has been realized by using a proprietary data mesh format. Data coming from different data sources can be easily converted to this format via linked readers.

VI. Application Examples

In the following, we briefly sketch two examples of interactive visual flow analysis, using our time-dependent feature specification setup. Further results, as well as a short video, demonstrating the interactive behaviour of the framework and the process of how to specify complex, time-dependent features, are available on the project home page [18]. Another, also more detailed application description is available in a case study report on the interactive visual analysis of a Diesel exhaust system [9]. In this case study, a subset of the time-dependent features which are discussed in this paper already proved to be very useful in a real-world analysis.

A. Flood after the burst of a dam

The first example demonstrates the investigation of a CFD simulation of the burst of a dam and the resulting flooding of the surrounding area. The dataset, which can be seen in figures 1 and 12, is made up of a grid of 76,505 cells, exhibits 28 attribute dimensions, and is organized in 48 time steps. The flood is simulated as a two-phase flow, comprised of water and air. The interesting region in this simulation is the region around an obstacle which is located closely to the bursting dam (on the right side in figure 14). In figure 12, a time-dependent feature has been specified to investigate how long the flood actually impacts this object. Also the height-level of the impact in different phases of the simulation can be clearly seen from this visualization.

The feature has been specified by smoothly brushing values of high cell fraction of water (high in cells where water is). The specification has then been refined by restricting the selection to represent only cells which stay at a high level of water cell fraction longer than just a few time steps (using the stationary attribute derivation of water cell fraction). The latter restriction eliminates all cells, where just for one or two time steps transient flooding occurs, i.e., we focus only on those parts which are seriously affected by the flood. The features in focus are colored according to the water cell fraction values, yellow denoting medium cell fractions (partly mixed with air) and red showing regions of water only. The temporal evolution in this figure is from upper left to lower right image.

What can be clearly seen is the recirculation zone in front of the obstacle. It is also interesting that the simulation yields a long-term high level of flooding. For details on the setup of the feature and an animation of the resulting feature, please refer to the accompanying video [18].

SimVis also has been connected to RTVR [22], which provides real-time volume rendering of 3D data – volume rendering means that a semi-transparent 3D data continuum is considered, which is rendered back-to-front through a large number of successive blending steps, eventually generating an impression of a semi-transparent data medium. So we are also able to visualize features as specified with our framework with high-speed, real-time volume rendering [16]. As the RTVR allows only rendering of regular grids, the exchange of data between SimVis and RTVR works via on-demand hardware-software balanced resampling of the DOI values into a regular grid, based on an algorithm by Weiler and Ertl [33].

Figure 14 shows an RTVR volume rendering of a feature describing areas of high water cell fraction at time $t = 38\,s$. The boundary surface of the dataset is rendered as a set of contour lines which reduces visual clutter due to occlusion in 3D rendering.

In figure 1 another time-dependent feature is visualized for three different time steps of this dataset (12, 20, and 46 seconds after simulation start). This feature shows the local maxima (red) and minima (green) for regions with significant volume fraction of water. As can be clearly seen, flow fronts are usually preceded/succeeded by alternating minima/maxima of volume fraction values, respectively.

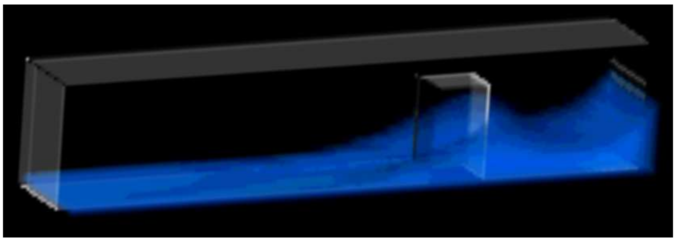

Fig. 14. Showing large amounts of water in the flood simulation data (dyke breach) by the use of volume rendering.

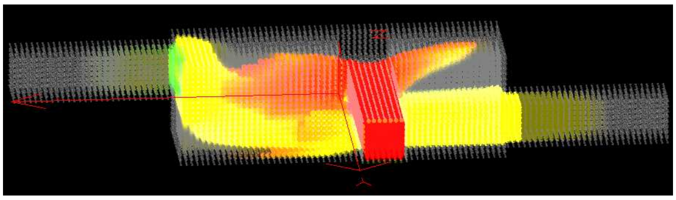

Fig. 15. Joint 3D rendering of all temperature data for all 100 time steps, highlighting high velocities. More opaque data subsets correspond to areas which remain in focus, i.e., which exhibit larger velocities, for a longer time.

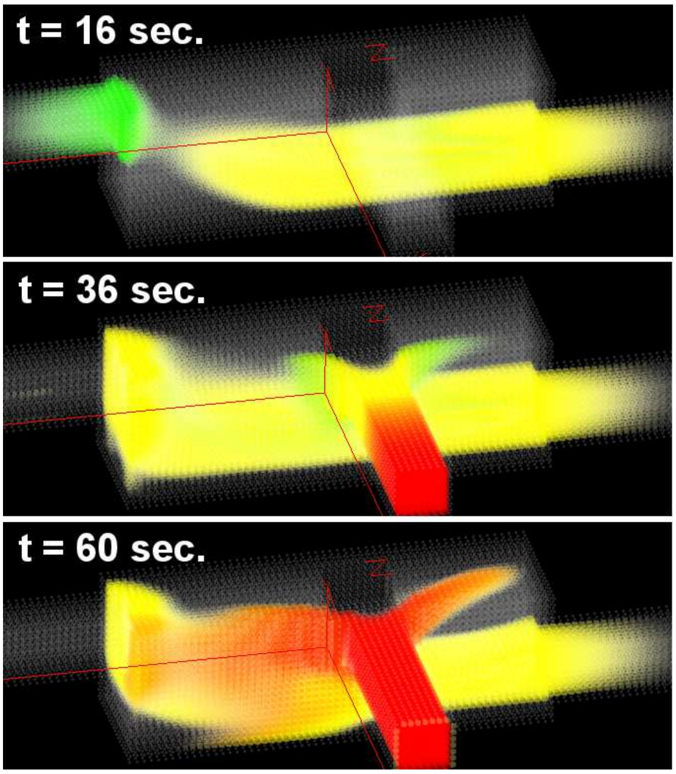

Fig. 16. Time steps 16, 36, and 60 of the extended T-junction dataset, showing the same feature as in figure 15, but time step by time step.

B. Mixing time-shifted flows in an extended T-junction

Figure 16 shows a second application example, namely an extended T-junction, consisting of two inlets (one on a lower level, right side, the other on a higher level, front) and one outlet (left side, upper level). Near the front inlet, an obstacle is blocking the flow. This dataset consists of 30,930 cells on an unstructured grid, each cell having 18 data attributes. The temporal domain spans 100 equidistant time steps.

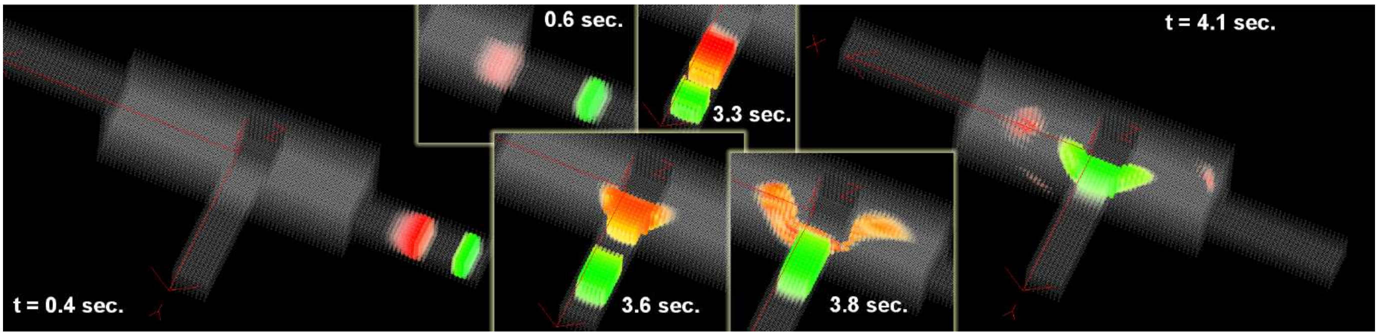

Fig. 17. The regions around two different flow fronts are selected by specifying a single time-dependent feature using first- and second-order derivatives of temperature values (see text for details).

The dataset has two time-shifted inflows from the two inlets, one starting at time step 0 from the right inlet, injecting warm fluid into the T-junction, the other one from the front inlet, starting at time step 33, injecting hot flow. The two flows mix in the chamber around the obstacle.

Figure 15 shows a 3D rendering of temperature values for the data of all 100 time steps of the dataset. The visualized feature is specified by high velocity values smoothly brushed for the middle part of the dataset (right inlet and left outlet have been neglected). The image shows, which parts are in focus and for how long (the more opaque, the longer in focus). The color codes global temperature values over the whole data, with red coding high values and green showing low temperature (initial temperature in the whole dataset).

When comparing this visualization to the three renderings of the data from single time steps in figure 16, the following difference can be observed: the 3D rendering of data from a longer time interval gives a general overview, which regions are in focus for longer periods of time. It smoothes out regions being in short-term focus, like the green, low temperature area in the upper image of time step 16 shown in figure 16. Figure 16 shows time steps 16, 36, and 60 using the same feature specification as in figure 15.

Another time-dependent feature is presented in figure 17. It represents regions, where the temperature derivatives are high and the rate of change of derivatives (second-order differentiation) is either very high, or very low. This shows regions, where temperature changes very much, but the change itself is only short-term. This corresponds to the front and backside of the flow fronts. By color coding according to the change of derivatives, the distinction between positive and negative changes of derivatives becomes visible in the 3D view. At time steps 0.4 s and 0.6 s after the start of the simulation, the first flow front shows up, whereas at time steps 3.3 s through 4.1 s the second flow front is visualized. Note, that this is captured by the specification of one feature (not temporally or spatially connected), although the two flow fronts represented by this feature exhibit different temperature ranges.

VII. Conclusions and Future Work

In this paper, we presented recent extensions to our SimVis framework for the interactive visual analysis of CFD simulation data, especially focussing on the demands of *time-dependent CFD data*. Extensions to the feature specification process (for the definition of time-dependent features) are realized through *attribute derivation* and *extended brushing* mechanisms. For the feature-based focus+context visualization, extensions with respect to the visualization of time-dependent flow data are presented. Performance improvements to the SimVis system, which became necessary to handle large datasets, are described.

Given a set of flexible tools for the interactive visual exploration and analysis of large simulation datasets, the user gains improved insight into CFD data. Attribute derivation, as means to enable effective access to information which inherently is dependent on time, as well as the support of data-range-independent, relative data assessment are new tools which support the specification of time-dependent flow features in time-dependent datasets.

Using focus+context visualization proves to be especially suitable for the intuitive visualization of large, multi-dimensional, and multi-variate datasets from unsteady CFD simulations. We extended conventional 3D rendering of data from singular time steps, by enabling a 3D visualization of data from longer time intervals containing multiple time steps to show temporal evolution of time-dependent feature characteristics in a single image.

We are currently working on the extension of the SimVis system to handle datasets which are based on time-varying grids during the analysis of time-dependent simulation data. First results have already been achieved (see figure 18 for an example of a visualization of data from four different time steps from the simulation of parts of the combustion chamber of a Diesel engine [8]). Future work will include the extension of the here presented attribute derivation methods for specifying time-dependent features to cope with the special challenges of data based on time-varying grids. The usage of position-based methods for feature extraction will have to be included, as opposed to cell-based methods. An example would be to calculate derivatives for absolute or relative spatial positions rather than cell-based temporal derivatives. Future work will also deal with the inclusion of explicit feature representations of specified features. This will enable comparisons of different feature specifications, and we thereby also plan to establish a link to previous feature extraction-

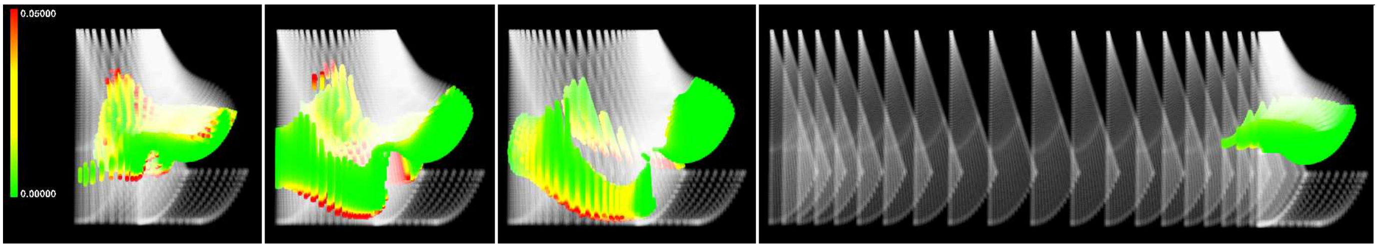

Fig. 18. Coping with datasets which are computed on time-varying grids: data from the simulation of parts of the combustion chamber of a Diesel engine application is shown for four different time steps side by side.

based methods.

ACKNOWLEDGEMENTS

This work has been carried out as part of the basic research on visualization at the VRVis Research Center in Vienna, Austria (http://www.VRVis.at/vis/), which is partly funded by an Austrian research program called Kplus. All CFD simulation datasets presented in this paper are courtesy of AVL List GmbH, Graz, Austria. The authors would like to thank Michael Mayer for helping with the data conversion and answering questions about the shown datasets.

Special gratitude goes also to our collegues Markus Hadwiger, who helped with parts of the implementation of the underlying mesh-library, and Matej Mlejnek, who developed the RTVR-based 3D rendering view for our framework. Furthermore the authors would like to thank Robert S. Laramee for his help in preparing this paper as well as the reviewers of this paper for their useful suggestions.

REFERENCES

[1] Hyper-threading technology architecture and microarchitecture. See URL: `http://www.intel.com/technology/itj/2002/volume06issue01/vol6iss1_hyper_threading_technology.pdf`.

[2] Webpage about SimVis. See URL: `http://www.SimVis.at/`.

[3] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 110–119, 2000.

[4] D. Bauer, R. Peikert, M. Sato, and Mirjam Sick. A case study in selective visualization of unsteady 3D flow. In *IEEE Visualization 2002*, pages 525–528.

[5] J. Clyne and J. Dennis. Interactive direct volume rendering of time-varying data. In *Data Visualization '99*, Eurographics, pages 109–120.

[6] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proc. of the 5th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 239–248, Grenoble, France, May 2003.

[7] H. Doleisch and H. Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. *Journal of WSCG*, 10(1):147–154.

[8] H. Doleisch, M. Mayer, M. Gasser, P. Priesching, and H. Hauser. Interactive feature specification for simulation data on time-varying grids. In *Proc. of the Conference Simulation and Visualization (SimVis 2005)*, pages 291–304, Magdeburg, Germany, 2005.

[9] H. Doleisch, M. Mayer, M. Gasser, R. Wanker, and H. Hauser. Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In *Proc. of the 6th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2004)*, pages 91–96, Konstanz, Germany, May 2004.

[10] Intel Streaming SIMD Extensions. See URL: `http://x86.ddj.com/articles/sse_pt1/simd1.htm`.

[11] G. Furnas. Generalized fisheye views. In *Proc. of the ACM CHI '86 Conf. on Human Factors in Computing Systems*, pages 16–23, 1986.

[12] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1995.

[13] M. Gasser. Fast focus+context visualization of large scientific data. Technical Report TR-VRVis-2004-005, VRVis Research Center, 2004.

[14] D. Gresh, B. Rogowitz, R. Winslow, D. Scollan, and C. Yung. WEAVE: A system for visually linking 3D and statistical visualizations, applied to cardiac simulation and measurement data. In *IEEE Visualization 2000*, pages 489–492.

[15] H. Hauser. Generalizing focus+context visualization. In *Scientific Visualization: The Visual Extraction of Knowledge from Data (Proceedings of the Dagstuhl 2003 Seminar on Scientific Visualization)*, pages 305–327, 2005.

[16] H. Hauser and M. Mlejnek. Interactive volume visualization of complex flow semantics. In *8th Fall Workshop on Vision, Modeling and Visualization (VMV 2003)*, pages 191–198.

[17] C. Henze. Feature detection in linked derived spaces. In *IEEE Visualization '98*, pages 87–94.

[18] Project home page of presented work. See URL: `http://www.VRVis.at/vis/research/simvis-time/`.

[19] D. Kenwright and R. Haimes. Automatic vortex core detection. *IEEE Computer Graphics & Applications*, 18(4):70–74, 1998.

[20] R. Kosara, F. Bendix, and H. Hauser. Timehistograms for large, time-dependent data. In *Proc. of the 6th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2004)*, pages 45–54, Konstanz, Germany, May 2004.

[21] E. Lum, K.-L. Ma, and J. Clyne. A hardware-assisted scalable solution for interactive volume rendering of time-varying data. *IEEE Trans. on Visualization and Computer Graphics*, 8(3):286–301, 2002.

[22] L. Mroz and H. Hauser. RTVR - a flexible java library for interactive volume rendering. In *IEEE Visualization 2001*, pages 279–286.

[23] R. Peikert and M. Roth. The parallel vectors operator - A vector field visualization primitive. In *IEEE Visualization '99*, pages 263–270.

[24] H. Piringer, R. Kosara, and H. Hauser. Interactive focus+context visualization with linked 2D/3D scatterplots. In *Proc. of the Intl. Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2004)*, London, UK, july 2004.

[25] F. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. Feature extraction and visualization of flow fields. In *Eurographics 2002 State-of-the-Art Reports*, pages 69–100, Saarbrücken Germany, September 2002.

[26] F. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. The state of the art in flow visualization: Feature extraction and tracking. *Computer Graphics Forum*, 22(2):775–792, Dec. 2003.

[27] F. Reinders. *Feature-Based Visualization of Time-Dependent Data*. PhD thesis, Delft University of Technology, The Netherlands, March 2001.

[28] M. Roth and R. Peikert. A higher-order method for finding vortex core lines. In *IEEE Visualization '98*, pages 143–150.

[29] D. Silver and X. Wang. Tracking features in unstructured datasets. In *IEEE Visualization '98*, pages 79–86.

[30] J. Swan, M. Lanzagorta, D. Maxwell, E. Kou, J. Uhlmann, W. Anderson, H. Shyu, and W. Smith. A computational steering system for studying microwave interactions with missile bodies. In *IEEE Visualization 2000*, pages 441–444.

[31] L. Treinish. Multi-resolution visualization techniques for nested weather models. In *IEEE Visualization 2002*, pages 513–516.

[32] M. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data. In *IEEE Visualization '94*, pages 326–336.

[33] M. Weiler and T. Ertl. Hardware-software-balanced resampling for the interactive visualization of unstructured grids. In *IEEE Visualization 2001*, pages 199–206.