A Four-level Focus+Context Approach to Interactive Visual Analysis of Temporal Features in Large Scientific Data

Philipp Muigg[1], Johannes Kehrer[1,3], Steffen Oeltze[2], Harald Piringer[1], Helmut Doleisch[1], Bernhard Preim[2], and Helwig Hauser[3]

[1] VRVis Research Center, Vienna, Austria
[2] Department of Simulation and Graphics, University of Magdeburg, Magdeburg, Germany
[3] Department of Informatics, University of Bergen, Bergen, Norway

Abstract

In this paper we present a new approach to the interactive visual analysis of time-dependent scientific data – both from measurements as well as from computational simulation – by visualizing a scalar function over time for each of tenthousands or even millions of sample points. In order to cope with overdrawing and cluttering, we introduce a new four-level method of focus+context visualization. Based on a setting of coordinated, multiple views (with linking and brushing), we integrate three different kinds of focus and also the context in every single view. Per data item we use three values (from the unit interval each) to represent to which degree the data item is part of the respective focus level. We present a color compositing scheme which is capable of expressing all three values in a meaningful way, taking semantics and their relations amongst each other (in the context of our multiple linked view setup) into account. Furthermore, we present additional image-based postprocessing methods to enhance the visualization of large sets of function graphs, including a texture-based technique based on line integral convolution (LIC). We also propose advanced brushing techniques which are specific to the time-dependent nature of the data (in order to brush patterns over time more efficiently). We demonstrate the usefulness of the new approach in the context of medical perfusion data.

1. Introduction

One interesting development of recent times is the increased share of time-varying scientific measurements and computations. In numerical simulation, time-dependent representations of dynamic phenomena are commonly used to solve real world problems. Additionally, fast imaging devices and improvements in the fields of registration and motion correction allow for time-dependent data also in the medical domain. Traditionally, time-dependent data is either visualized by displaying separate individual time steps or by specifying a few positions within the dataset for which the evolution of a data attribute is visualized over time (by drawing function graphs). Both approaches provide only a selective view of the dataset, either with respect to time, when showing just one timestep at a time, or with respect to space, when showing the evolution of an attribute for a few sample positions, only. Users interested in temporal features in a time-dependent dataset either have to mentally integrate multiple images from different time steps or search for interesting spatial regions by choosing different sample locations and looking at the resulting function graphs. Especially if the temporal evolution of corresponding data items, e.g., the change of temperature within a Computational Fluid Dynamics (CFD) cell over time, is investigated, the latter, curve-based approach is far more effective than looking at multiple snapshots or animations. Still the selection of regions of interest, especially during data exploration and analysis, is a tedious task if no proper context information about the overall dataset is provided.

We propose a new dense visualization approach by displaying function graphs for all cells/voxels of a volumetric, time-dependent dataset. This leads to visualizations consisting of hundreds of thousands or even of millions of function graphs, sampled at a relatively low number of time steps. Existing visualization technology for time series data has not yet addressed such large amounts of function graphs con-
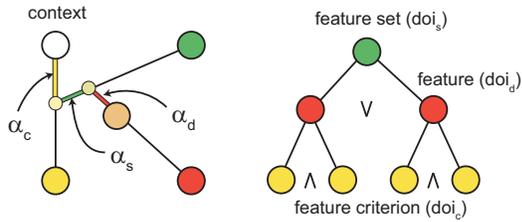
Figure 1: *The structure of the FDL tree (right) and the optimized blending scheme for color coding (left). The three colors represent different levels of the FDL tree.*

currently. In this paper we demonstrate our solution for this challenging problem.

In order to improve on occlusion and cluttering problems, resulting from the huge amount of function graphs to draw, we utilize a highly interactive four-level focus+context method. We use data aggregation and image space methods to retain responsiveness even when interacting with large datasets. Our approach is integrated with an interactive visual analysis approach of coordinated multiple views [DGH03]. Features are specified by the user in a hierarchical scheme which is based on individual selections and logical combinations of them on higher levels. We allow for smooth transitions between focus and context, based on the concept of smooth brushing [DH02], which are represented by degree of interest (DOI) attributions of the data. We consider three different DOI values per data item and present how this is integrated with our visual analysis approach.

In order to account for the temporal nature of the data, we extend the brushing functionality of our framework by *function-similarity brushes* which are based on a user definable pattern. These patterns, which can be sketched and modified interactively, are matched to the data curves, also resulting in smooth DOI attributions. Graphical context information is provided by using line integral convolution [CL93] (LIC) on a synthetic flow field which represents the temporal evolution of each data item. We also propose to use a binned data representation to increase the rendering performance as well as additional image-based techniques to further enhance the visualization.

2. Related Work

A large variety of publications deal with the visualization and analysis of *time-oriented information* (see the work of Aigner et al. [AMM*08] for an overview). Several applications exist that allow the analysis of time-dependent data using interactive brushing or querying techniques. Feature visualization and specification via linking and brushing in multiple views is an integral part of the *SimVis* framework [Dol04]. Scalar degree of interest (DOI) values from the unit interval represent the membership with respect to features for all data items. The values are assigned, for example, using *smooth brushing* [DH02], where a linear border region is assumed between focus (DOI = 1) and context

(DOI = 0), and focus+context visualization is applied to represent features.

In order to reduce visual cluttering, Johansson et al. [JLJC06] use high-precision *density maps* [WL97] in their work on parallel coordinates that count the number of primitives (*density*) that pass though each screen pixel. User-defined *transfer functions* are applied that map density to opacity values in the final output. However, the rendering of the individual primitives into the density map can take up to a few seconds when visualizing large data sets. Novotný and Hauser [NH06] apply 2D *bin maps* as data representation mechanism in parallel coordinates in combination with outlier detection and clustering within the maps. Depicting the binned information allows the focus+context visualization of large data sets showing data trends while preserving outliers at interactive frame rates.

Konyha et al. [KMG*06] introduce *line brushes* to select a subset of function graphs out of a large number of graphs, which intersect with a simple line segment drawn in the view. The *TimeSearcher* [HS04] is an application especially designed for the visual analysis of time series using *Time Boxes* or *angular query widgets*. The latter are used to select time series which have a similar slope on a sequence of time steps (compare to *angular brushing* [HLD02]). Further extensions of the TimeSearcher [BAP*05] allow for *similarity-based querying* of temporal patterns. *QuerySketch* [Wat01] enables the user to outline the shape of a pattern used for querying directly in the view. Inspired by this technique, *QueryLines* [RLL*05] allows the graphical specification of approximate queries, where soft constraints and preferences are used for fuzzy pattern description and ranking of the query results, respectively.

Our approach combines several of the previously mentioned methods in a novel way and introduces additional functionality for the definition and visualization of temporal features using multiple linked views.
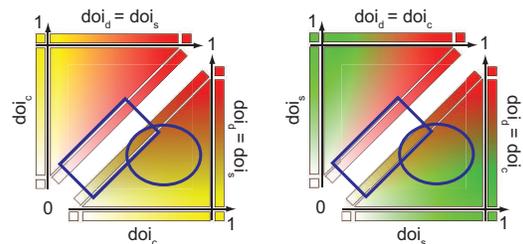

Figure 2: *Two different color blending approaches (combining red representing* doi_d, *green* doi_s, *and yellow* doi_c*) are illustrated. The top-left triangle of each side shows our compositing scheme using* α_s, α_d, *and* α_c *whereas the lower-right parts apply standard alpha compositing using* doi_s, doi_d, *and* doi_c *as weights. Note that especially along the diagonal and the borders a brownish tint is introduced by standard alpha compositing because of color mixing.*
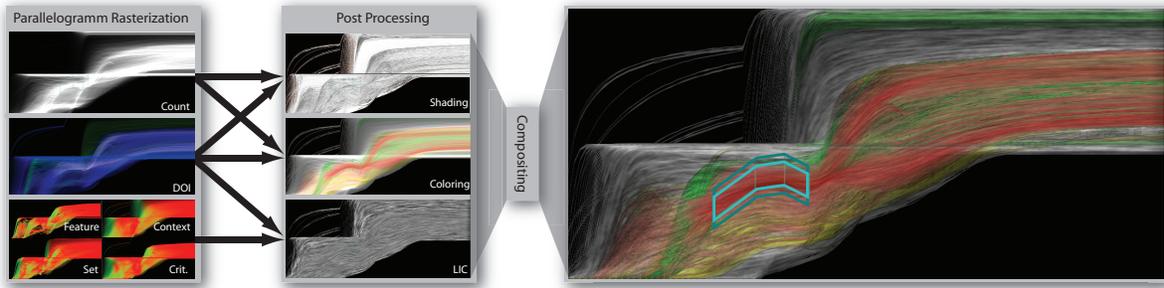
Figure 3: *The first stage of our rendering pipeline rasterizes parallelograms (each representing multiple function graph segments) and generates multiple output images which are used by the post processing stage. Here different image based algorithms are applied in order to create three different effects (shading, coloring, and LIC), which are combined in the compositing stage.*

3. Four-Levels of Focus+Context

Generally a *feature* within a dataset can be defined as a subset of data items which meet multiple criteria specified by the user. The framework, into which our approach has been incorporated, provides different linked views (such as histograms, scatterplots or the proposed function graph visualization) that can be used to specify a DOI function representing one such *feature criterion*. In order to specify multiple features, which themselves are defined by multiple criteria, we use a Feature Definition Language (FDL) tree [DGH03] (see figure 1 (right)). It consists of three levels and is organized as disjunctive normal form. The DOI functions of leaf nodes, the *feature criteria*, are combined into DOI functions representing *features* by applying a fuzzy "AND" operation (all criteria have to be valid for a data item in order to belong to a feature). *Features* themselves are combined into *feature sets* by using a fuzzy "OR" operation.

In order to analyze the influence of the criterion specified within one view on its containing feature and feature set we propose to visualize the respective DOI functions. Figure 3 (right) shows an example where CFD simulation data has been used. For each cell temperature is plotted over time. Coloring is used to visualize whether a data item is selected only in the current view (yellow), in the containing feature (red), in the containing feature set (green) or not selected at all (grey). The problem of properly blending these three smooth DOI functions with a context representation is tackled by our four-level compositing scheme (i.e., three focus plus one context level) which is discussed in the context of the proposed function graph visualization in the remainder of this section.

3.1. The Four-Level Compositing Scheme

The three different DOI functions which have to be visualized for each data item in the context of a current view are semantically related to each other: The DOI representing a feature criterion (doi_c) directly reflects selections within the current view. No other parts of the FDL tree (related to other views) are considered. The DOI within a feature (doi_d) is always less than or equal to doi_c (since doi_d is an AND com-

bination of doi_c values as shown in figure 1). Contrarily, the DOI of the feature set (doi_s) is always greater or equal than every contained doi_d (due to the OR combination). In order to properly express the contribution of each DOI, we order them based on their influence. The doi_s is the top level collection of individual features defined in their respective doi_d. Thus doi_d is visualized above doi_s. This allows the user to discriminate between the feature containing the current view and all remaining features in the common feature set (compare red and green regions in figure 3 (right)). In order to provide additional information about the feature criterion defined in the current view doi_c is visualized behind doi_d and doi_s.

If colors are blended together to represent the different smooth DOIs it is crucial that as few color mixing as possible occurs in order to allow for straight forward interpretation [Bre99]. Thus simple alpha blending, using the different DOIs as coefficients, has not been used (the color mixing resulting from directly using the DOIs as weights is illustrated in the two lower-right triangles in figure 2). Instead we use repeated alpha blending with the following weights:

$$\alpha_c = \max(doi_c - doi_s, 0) \quad \alpha_s = doi_s - doi_d \quad \alpha_d = doi_d \quad (1)$$

These weights avoid the introduction of additional tints as shown in the upper-left triangles in figure 2 because they avoid adding colors if they represent only redundant information. For example adding the color assigned to the containing feature set if its DOI function is equal to the current feature's is redundant (no other features contribute to the definition of the feature set). Thus we focus on the difference between doi_s and doi_d which indicates that other features than the current one contribute to the containing feature set.

The order in which the alpha blending is performed is based on the already mentioned influence of each DOI resulting in blending doi_d over doi_s over doi_c over a context representation, each using their respective alpha value (α_c, α_s, α_d). This is illustrated in figure 1 (left) using colors which are applied throughout the remainder of this paper.

3.2. Visualizing Binned Curves

Since the SimVis framework is dealing with relatively large datasets (e.g. 500K time series) we are using an aggregated data representation to interactively visualize a function over time for each sample in our function graph visualization. As proposed by Novotný and Hauser [NH06] in the context of parallel coordinates we split the time series data into segments between two consecutive time steps. Each of these segments (see figure 4 left) can be represented by a *frequency binmap* (see figure 4 right) which basically is a 2D histogram. The data values of the two adjacent time steps are used as axes and each bin stores the number of function graphs passing through. It is important to note that the approach proposed by Novotný and Hauser did only use one binmap to store frequency information. Since we have to visualize three additional scalar values we introduce additional *DOI binmaps*. Because the DOI functions can vary over time it is necessary to use six additional DOI binmaps per segment: two for each of the three DOI functions, representing their aggregated values at the left and the right timestep of the segment.

In order to properly reconstruct the original curves and to apply the proposed four-level focus+context compositing method to coloring and an adapted line integral convolution (LIC) approach, we use the rendering pipeline presented in figure 3.

Parallelogram Rasterization: In this first pipeline stage the frequency representation stored in the individual binmaps is translated back into the time/attribute domain. We rasterize one parallelogram for each bin of the binmaps which contains at least one entry (see figure 4 for the underlying geometry). Since the data from two consecutive time steps is often highly correlated only a small fraction of the bins has to be visualized (for example only 2.1% of the 256×256 bins per time segment have to be displayed for the simulation dataset shown in figure 3).

Each such parallelogram has to represent different properties of the underlying frequency data which are accumulated per image pixel by additive blending and multiple floating point precision render targets. These properties include the number of function graphs c within a bin as shown in sub-figure "count" of figure 3. High luminance indicates a large
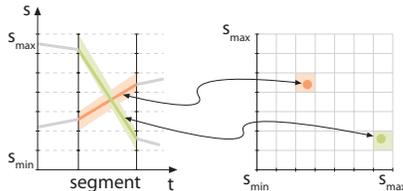

Figure 4: *This figure illustrates the duality between bins in a binmap and function graph segments in the corresponding function graph space.*

number of function graphs passing through a pixel. Additional properties are the averaged DOI values multiplied by c in order to account for the fact that each parallelogram represents a different number of function graph segments. The averaged DOI values are fetched from the DOI binmaps and interpolated linearly across the parallelograms. This enables us to visualize the DOI (which often varies over time) for individual sample points along the function graph. Sub-figure "DOI" of figure 3 shows the accumulated DOIs (each color channel representing one of doi_c, doi_d or doi_s).

Besides the frequency and DOI information we propose to include the temporal evolution of the data into our visualization by applying a flow visualization technique (LIC) to a vector field which represents temporal trends. This vector field is created by accumulating the normalized direction vectors $\mathbf{d} = |(y_{i+1} - y_i, t_{i+1} - t_i)|$ of the parallelograms per image pixel. Here y_i and y_{i+1} are the scalar data values represented by the parallelogram at timestep i and $i+1$ and t_i and t_{i+1} are the corresponding points in time. As mentioned previously we accumulate $\mathbf{d} \cdot c$ in order to account for the number of function graphs represented by a parallelogram. Since we treat the three different DOI attributes and the context separately we propose to additionally accumulate $\mathbf{d} \cdot doi_s \cdot c$, $\mathbf{d} \cdot doi_d \cdot c$, and $\mathbf{d} \cdot doi_c \cdot c$ as shown in sub-figures "Context", "Set", "Feature", and "Crit." of figure 3 (here the direction is mapped to the red and green color channels). This allows us to blend between these four different vector fields when applying LIC in order to visualize the temporal evolution of the overall data and the selected subsets represented by the DOI functions separately.

The results of this first stage are eight images containing accumulated information about all function graphs: $C(x,y)$ the number of function graphs through pixel (x,y), $DOI_s(x,y)$, $DOI_d(x,y)$, and $DOI_c(x,y)$ the average DOI of all function graphs, and $\mathbf{D}(x,y)$, $\mathbf{D}_s(x,y)$, $\mathbf{D}_d(x,y)$, and $\mathbf{D}_c(x,y)$ storing the average direction of all function graphs through pixel (x,y) weighted by the respective DOI.

Post Processing: The post-processing stage first performs simple transformations on some images: $\overline{C}(x,y) = C(x,y) \cdot s + t$, $\overline{DOI}_s(x,y) = DOI_s(x,y)^\gamma$, $\overline{DOI}_d(x,y) = DOI_d(x,y)^\gamma$, and $\overline{DOI}_c(x,y) = DOI_c(x,y)^\gamma$. The scale factor s and offset t can be used to linearly modify the brightness of the result visualizations whereas γ is used to enhance the contrast of the DOI attributations in order to highlight selected outliers.

Based on these transformed images three different outputs are computed which are weighted and combined in the compositing stage. First the images $\overline{C}(x,y)$, $\overline{DOI}_s(x,y)$, $\overline{DOI}_d(x,y)$, and $\overline{DOI}_c(x,y)$ are interpreted as height fields. Thus a diffuse lighting contribution can be computed using a directional light source above the height fields. The first output image, as depicted in figure 5 (a), represents the minimum of all four lighting contributions. This is similar to basic edge enhancement which allows the user to recognize sharp boundaries of trends in the four underlying images.

The consecutive outputs are based on our four-level focus+context compositing scheme. Figure 5 (b) shows the result of applying the compositing to four color values as already indicated in figure 1. Additionally the result color is modulated by $\overline{C}(x,y)$ in order to account for function graph density. The final output shown in figure 5 (c) is also based on our compositing approach. But instead of compositing colors on a pixel per pixel basis we blend the different direction fields $\mathbf{D}(x,y)$, $\mathbf{D}_s(x,y)$, $\mathbf{D}_d(x,y)$, and $\mathbf{D}_c(x,y)$ using the corresponding (transformed) DOI images. This synthetic vector field represents the overall evolution of all function graphs over time with special focus on the three DOI layers (thus trends in selected regions always override trends visible in the context). In order to visualize this field we apply line integral convolution by sampling from a low and a high frequency noise texture along Euler interpolated streamlets. The final values which are accumulated along the streamlets are a linear combination of the high and low frequency samples (s_h and s_l) based on $\lambda = \max(\overline{\mathrm{DOI}}_s(x,y), \overline{\mathrm{DOI}}_d(x,y), \overline{\mathrm{DOI}}_c(x,y))$:

$$s = s_h \cdot \lambda + s_l(1-\lambda) \qquad (2)$$

Thus regions with high DOI values are displayed using finer noise whereas low DOI regions are represented by coarser noise as shown in figure 5 (c).

Compositing: As implied by its name this step combines the three images (shading i_s, coloring i_c, and LIC i_l) based on three weights (w_s, w_c, and w_l respectively) resulting in a visualization as shown in figure 3. The blending is performed according to this equation:

$$(i_s w_s + (1-w_s)) \cdot (i_c w_c + (1-w_c)) \cdot (i_l w_l + (1-w_l)) \quad (3)$$

We chose to modulate all images with each other in order to retain a maximum amount of contrast whereas the different weights allow for an easy adjustment of the influence of single post processing effects. Since only the contribution from i_c contains color information (i_s and i_l contain luminance values) the resulting color is just attenuated (not distorted).
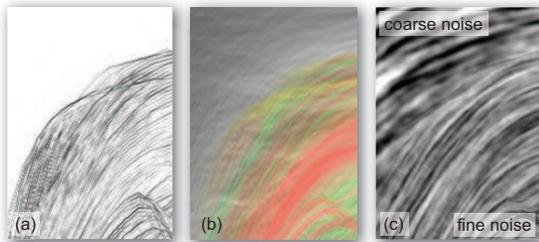
Figure 5: *Cutout images from figure 3. Shading (a) is used to emphasize "edges" in the DOI and frequency images whereas the application of our compositing scheme translates the three different DOI attributations into easily recognizable colors (b). Additionally LIC is applied to a flow field representing the average function graph directions (c).*
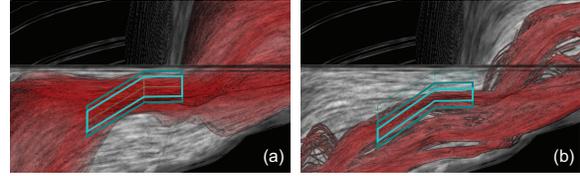
Figure 6: *This figure shows brushing results based on two different similarity measures. In (a) no data transformation is used whereas in (b) a gradient based distance is evaluated which is invariant to vertical translation.*

The result images in this paper have been generated with settings of $w_s = 0.7$, $w_c = 1$, and $w_l = 0.7$.

3.3. Brushing

When analyzing a large number of function graphs it is often necessary to compare them against a certain pattern. For example, the user may be interested in functions showing a sudden rise at some point in time with subsequent stabilization. Such queries can to some degree be approximated by combining multiple vertical interval selections at different positions within the domain (as also supported by our framework). However, brushing function graphs by roughly outlining a target function and evaluating the distances proved to be faster and more intuitive for many tasks and offers more options like considering average distances or ignoring vertical offsets. Basically, various ways of defining the target function are conceivable (e.g. analytical). In our case, the target is specified by linearly interpolating between an arbitrary number of control points (t_i, y_i) with t_i being a discrete time-step and y_i the according value in the data domain. After defining the target function, the basic steps generally necessary for evaluating the brush are a transformation of the sampled raw data, an aggregation of differences, and finally applying one (or two for smooth brushing) threshold(s) to obtain the degree-of-interest values.

As an (optional) first step, different transformations may be applied prior to evaluation. Without transformation, similarities are computed on the raw data values with respect to the absolute position of the target function (see figure 6 (a)). However, it is often desirable to account for task- and data-specific issues like offset translation, magnitude scaling, linear trends removal or noise reduction [KK02]. Disregarding constant offsets is particularly necessary in many cases, as the computed distances can otherwise become meaningless for measuring (subjective) similarity [KK02]. We propose two different ways of accounting for vertical offset translation (horizontal offsets are prohibitively expensive to determine for interactively brushing up to millions of function graphs): The first approach, shown in figure 6 (b), is based on computing the first derivative of the target function and all function graphs of the dataset, i.e., considering gradients rather than the raw data values themselves (e.g. estimated by forward differences or central differences). While this is comparatively fast because the transformation and the sub-

sequent aggregation can be done within one pass through the data, it is susceptible to noise. To account for this, the second approach seeks to align two functions on each other before comparing them by subtracting the average Y-value within the common interval $[T_1, T_2]$. For a function $f(t)$, this value is given by $(\int_{T_1}^{T_2} f(t)dt)/(T_2 - T_1)$. This approach is more stable, but may still be distorted by very strong "outliers" within the sample points and needs an own pass for computing the average value per function graph.

As the next step, the difference function $d(t) = |\hat{f}(t) - \hat{g}(t)|$ is computed for each transformed function graph $\hat{f}(t)$ with $\hat{g}(t)$ denoting the transformed goal function and aggregated within the common interval $[T_1, T_2]$. Among others, useful aggregations are the absolute maximum of $d(t)$ and the average distance $(\int_{T_1}^{T_2} d(t)dt)/(T_2 - T_1)$. Note that distances computed this way have the same scaling as the Y range and are independent of the scaling of the time domain. For function graphs given by discrete, unequally sampled points, it is important to weight the distances by the size of the according sample intervals when summing them up.

The final step is computing degree of interest (DOI) values from the distances by applying one (for binary brushing) or two (for smooth brushing) threshold(s). The latter case as used in our approach is illustrated in figure 7. As for the distance values, thresholds are specified in the value range of the function.

Interaction is very important for function-similarity brushes: After defining a target function by setting initial control points, further points can be added, existing ones can be modified or removed, and it is also possible to drag around the whole function. Adapting the thresholds is usually necessary and conveys a feeling at which gradient the similarity is changing for various regions. The evaluation is executed in parallel to interaction (i.e., threaded), which prevents disturbing delays and provides visual feedback as soon as possible.

4. Case Study Perfusion Data

Perfusion data are dynamic medical image data which characterize the regional blood flow in human tissue. They are acquired for example for tumor detection and classification in the female breast or to diagnose an acute ischemic stroke. We discuss two application examples based on datasets from Magnetic Resonance (MR) perfusion imaging. The distribution of contrast agents (CA) is registered to assess tissue kinetics (breast tumor diagnosis) and blood flow (ischemic
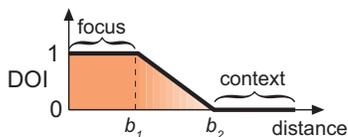
Figure 7: *Computing smooth DOI values based on distance values by linearly interpolating between two thresholds.*
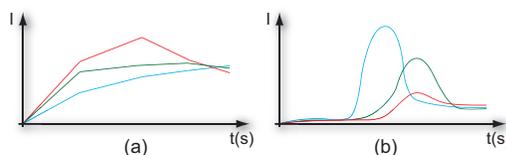
Figure 8: *(a) Time-intensity curves (TICs) typical for contrast agent accumulation in breast tissue. The continuous gentle inclination of the blue curve is characteristic for healthy tissue. The green and red curves are suspicious due to their high early enhancement. The red curve is especially suspicious because of its subsequent rapid wash-out, which is typical for malignant tumors. (b) TICs typical for contrast agent accumulation in the gray matter of the brain. The blue curve shows normal brain perfusion whereas the green and red curves show decreased and delayed perfusion around an infarction.*

stroke diagnosis). The CA is injected intravenously and its distribution is measured by a repeated acquisition of subsequent images covering the volume of interest. The CA provides signal changes in the acquired 4D data (3D+time). For each image voxel, a time-intensity curve (TIC) characterizes the CA enhancement (see figure 8). Clinicians are trained to infer tissue characteristics from the shape of these curves. The function graph visualization presented here allows the clinician to define a curve pattern and a similarity measure. Latter is applied to match the original TICs with this pattern. To our knowledge, this represents a new approach for the analysis of perfusion data. Existing approaches base the feature detection either on percental enhancement between two time-steps [CGB*05], or on so-called perfusion parameters derived from the TICs [ODH*07].

4.1. Breast Tumor Diagnosis

The process of CA enhancement in a tumor can be described by the diffusion of tracer particles from the inside of blood vessels into the extravascular space and vice versa before it becomes excreted in the kidneys. The permeability of the vessel walls and the extracellular volume fraction determine the amplitude and the shape of the TIC. TICs which show a high early enhancement (wash-in) followed by a rapid wash-out, i.e., a significant decrease of signal intensity afterwards, are especially suspicious (see figure 8 (a), red curve). Less suspicious are curves showing a plateau later on (green curve), or regions that continue to enhance (blue curve). This is typically observed in benign tumors. The major diagnostic task is to confirm or reject the hypothesis of a tumor being malignant. For more details on tumor perfusion, see Heywang-Köbrunner et al. [HKVHK97].

The dataset analyzed in the following was acquired to examine a suspicious region in the right mamma which had been detected during conventional mammography. As a preprocessing step, the original data has been cropped to restrict the analysis to relevant breast tissue. The resulting dataset matrix is: 458 x 204, slice distance: 3 mm, number of slices:
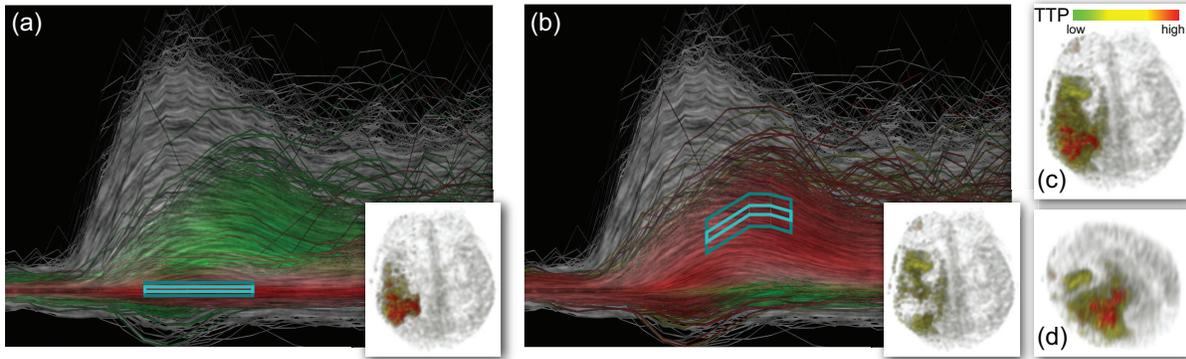
Figure 9: *A feature set containing two different features has been defined in two function graph visualizations. In (a), a smooth similarity brush (blue) has been defined such that tissue with no significant signal enhancement is selected in the first feature (shown in its 3D context in the inlet). In (b), a similarity brush has been used to specify a second feature representing tissue exhibiting reduced and delayed perfusion. Candidate areas of this tissue at risk are revealed around the core (inlet). In (c) and (d) the overall feature set is visualized. The color is modified according to the time it takes until the maximum amount of contrast agent is delivered for a particular voxel (TTP).*

26 (overall 2.429K voxels), temporal resolution: 6 measurements in 10 min. According to the accompanying diagnostic report, the patient suffered from a probably malignant structure (S_{large}) close to the thoracic wall and a smaller satellite lesion (S_{small}). The detection of the two structures and the verification of the reported findings are illustrated in figure 10. In order to select data items exhibiting a temporal pattern typical for malignant tissue a similarity brush using our gradient based measure is defined in a function graph view (a). As a result of this feature definition, S_{large} and S_{small} are revealed in a linked 3D visualization (b). The shape of the breast is indicated as context information.

4.2. Ischemic Stroke Diagnosis

In contrast to leaky vessels in malignant tumors, microvessels in normal brain tissue do not leak as a result of the blood brain barrier. Consequently, there is no enhancement in the extracellular volume. Instead, we observe the "first-pass" of the CA through the vessel components. In case of

Figure 10: *Detection of suspicious structures in breast tumor diagnosis. In our function graph visualization (a), a smooth similarity brush (blue) has been defined that resembles a curve pattern which is typical for malignant tumors. The resulting selection is visualized in the context of the entire right mamma in (b). Two suspicious structures S_{large} and S_{small} are revealed. The color indicates the accumulation speed during the early phase of the contrast agent passage (Slope).*

an ischemic stroke, the existence and the extent of "tissue at risk" surrounding the core of the stroke has to be evaluated immediately after the infarction. "Tissue at risk" is characterized by reduced and delayed perfusion (figure 8 (b)). Surgical and chemical interventions may salvage at least parts of the "tissue at risk" [dBF97].

The second dataset has been acquired immediately after an acute stroke. The dataset matrix is: 128 x 128, slice distance: 7 mm, number of slices: 12, temporal resolution: 40 measurements in 40 sec. According to the accompanying diagnostic report, the patient suffered from an infarction in the right hemisphere (which appears left in each view of figure 9). The detection of the infarction core and the surrounding "tissue at risk" is illustrated in figure 9. We have specified a feature set containing two features, the infarction core (a) and "tissue at risk" (b), by using two different function graph views and similarity brushes. Each view shows locally selected data (red) in combination with curves selected in the overall feature set (green). This allows for easy comparison of both features. Our LIC based approach helps to convey general temporal trends in the context (grey) as well as in the selected regions (colored). The inlets of (a) and (b) each show the respective feature in their spatial context. The color is modified according to the time it takes until the maximum amount of CA is delivered for a particular voxel, the so-called Time to Peak (TTP). Candidate areas for "tissue at risk" appear yellowish (medium TTP values) whereas the infarction core is colored red. In order to visualize the extent of the infarction, the overall feature set is shown in (c) and (d).
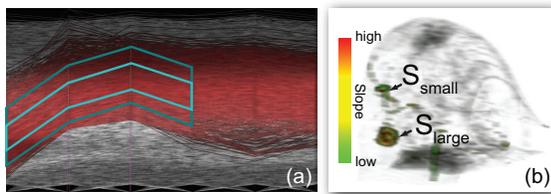
Our observations in this section could be successfully validated with results from Oeltze et al. [ODH*07] where the same datasets have been examined.

5. Conclusions and Future Work

We have presented an approach which effectively handles hundreds of thousands or even millions of function graphs interactively. A novel four-level focus+context method is used to blend both color and directional information according to the degree of interest specified in the various layers. The directional information is derived from the average orientation of function graphs through each pixel allowing for an effective visualization (by applying LIC) of large trends for the whole dataset as well as in regions selected by the user. We propose different smooth brushing techniques allowing for fast and intuitive selection of temporal trends based on user definable target functions. Additionally shading is used to indicate strong variation in the function graph density. In order to show the usefulness of our method, it has been successfully applied to medical perfusion data.

For future work we would like to abandon the necessity to explicitly rasterize a parallelogram for each bin within a binmap. Instead we plan to utilize the duality between a binmap and its corresponding segment by sampling along a line in the binmap to accumulate all bins which affect one pixel.

6. Acknowledgements

References

[AMM*08] AIGNER W., MIKSCH S., MÜLLER W., SCHUMANN H., TOMINSKI C.: Visual methods for analyzing time-oriented data. *IEEE TVCG 14*, 1 (2008), 47–60.

[BAP*05] BUONO P., ARIS A., PLAISANT C., KHELLA A., SHNEIDERMAN B.: Interactive pattern search in time series. In *Proc. IST/SPIE's 17th Ann. Intl. Symp. Electronic Imaging (VDA '05)* (2005), vol. 5669, pp. 175–186.

[Bre99] BREWER C. A.: Color use guidelines for data representation. In *Proc. of Section on Statistical Graphics, American Statistical Association, Alexandria VA.* (1999), pp. 55–60.

[CGB*05] COTO E., GRIMM S., BRUCKNER S., GRÖLLER E., KANITSAR A., RODRIGUEZ O.: MammoExplorer: An Advanced CAD Application for Breast DCE-MRI. In *Proc. Vision, Modelling, and Visualization (VMV)* (2005), pp. 91–98.

[CL93] CABRAL B., LEEDOM L. C.: Imaging vector fields using line integral convolution. In *Proc. of ACM SIGGRAPH '93* (1993), pp. 263–272.

[dBF97] DEN BOER J., FOLKERS P.: MR perfusion and diffusion imaging in ischaemic brain disease. *Medica Mundi 41*, 2 (1997), 20–35.

[DGH03] DOLEISCH H., GASSER M., HAUSER H.: 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)* (2003), pp. 239–248.

[DH02] DOLEISCH H., HAUSER H.: Smooth Brushing for Focus+Context Visualization of Simulation Data in 3D. In *Proc. of WSCG 2002* (2002), pp. 147–154.

[Dol04] DOLEISCH H.: *Visual Analysis of Complex Simulation Data using Multiple Heterogenous Views*. PhD thesis, Vienna University of Technology, Austria, 2004.

[HKVHK97] HEYWANG-KÖBRUNNER S., VIEHWEG P., HEINIG A., KUCHLER C.: Contrast-enhanced MRI of the breast: accuracy, value, controversies, solutions. 94–108.

[HLD02] HAUSER H., LEDERMANN F., DOLEISCH H.: Angular brushing of extended parallel coordinates. In *Proc. IEEE Symposium on Information Visualization 2002 (InfoVis 2002)* (2002), pp. 127–130.

[HS04] HOCHHEISER H., SHNEIDERMAN B.: Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization 3*, 1 (2004), 1–18.

[JLJC06] JOHANSSON J., LJUNG P., JERN M., COOPER M.: Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Information Visualization 5*, 2 (2006), 125–136.

[KK02] KEOGH E., KASETTY S.: On the need for time series data mining benchmarks: a survey and empirical demonstration. In *Proc. of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)* (2002).

[KMG*06] KONYHA Z., MATKOVIC K., GRACANIN D., JELOVIC M., HAUSER H.: Interactive visual analysis of families of function graphs. *IEEE TVCG 12*, 6 (2006), 1373–1385.

[NH06] NOVOTNY M., HAUSER H.: Outlier-preserving focus+context visualization in parallel coordinates. *IEEE TVCG 12*, 5 (2006), 893–900.

[ODH*07] OELTZE S., DOLEISCH H., HAUSER H., MUIGG P., PREIM B.: Interactive visual analysis of perfusion data. *IEEE TVCG 13*, 6 (2007), 1392–1399.

[RLL*05] RYALL K., LESH N., LANNING T., LEIGH D., MIYASHITA H., MAKINO S.: Querylines: approximate query for visual browsing. In *Extended abstracts on Human factors in computing systems (CHI '05)* (2005).

[Wat01] WATTENBERG M.: Sketching a graph to query a time-series database. In *Extended abstracts on Human factors in computing systems (CHI '01)* (2001), ACM Press, pp. 381–382.

[WL97] WEGMAN E. J., LUO Q.: High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics 28* (1997), 352–360.