Eurographics Conference on Visualization (EuroVis) 2016
K.-L. Ma, G. Santucci, and J. van Wijk
(Guest Editors)

Volume 35 (2016), Number 3

Time-Series Plots Integrated in Parallel-Coordinates Displays

Henning Gruendl[1], Patrick Riehmann[1], Yves Pausch[2], and Bernd Froehlich[1]

[1]Bauhaus-Universität Weimar, Virtual Reality and Visualization Research Group, Germany
[2]Dynardo GmbH, Weimar, Germany

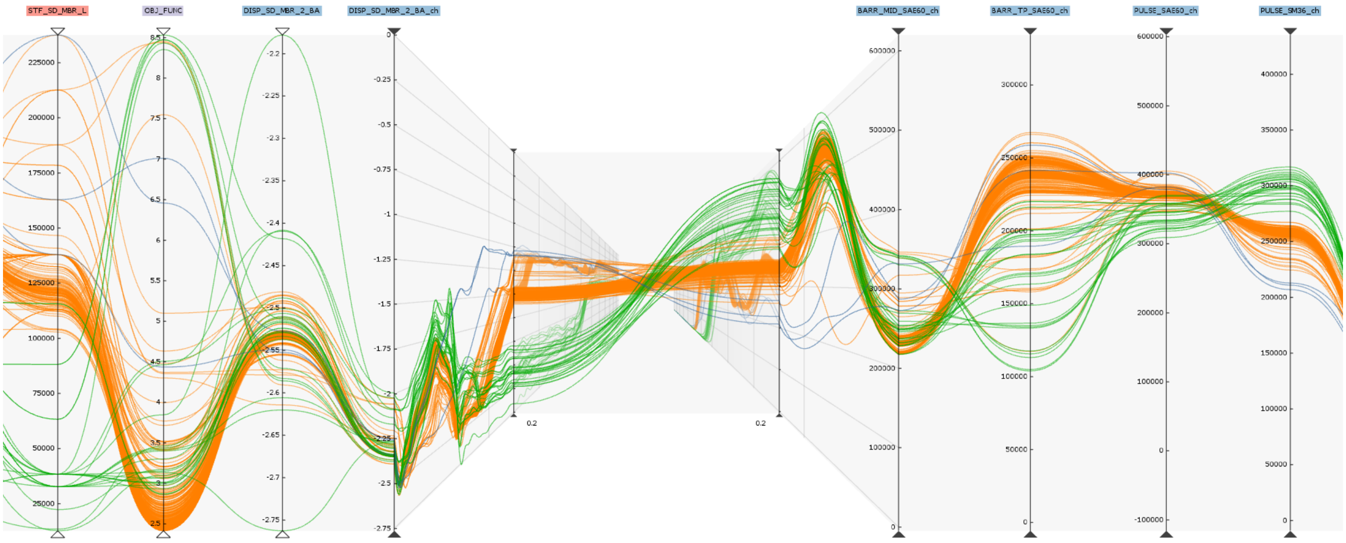

Figure 1: *A pseudo-perspective view of two time-series plots is integrated between two adjacent axes of a parallel-coordinates display. An independent translucent parallel-coordinates panel connects the two time-series plots and allows the analysis of trends in time as well as the exploration of changes in the relationship between the two time-dependent attributes.*

Abstract

We present a natural extension of two-dimensional parallel-coordinates plots for revealing relationships in time-dependent multi-attribute data by building on the idea that time can be considered as the third dimension. A time slice through the visualization represents a certain point in time and can be viewed as a regular parallel-coordinates display. A vertical slice through one of the axes of the parallel-coordinates display would show a time-series plot. For a focus-and-context integration of both views, we embed time-series plots between two adjacent axes of the parallel-coordinates plot. Both time-series plots are drawn using a pseudo three-dimensional perspective with a single vanishing point. An independent parallel-coordinates panel that connects the two perspectively displayed time-series plots can move forward and backward in time to reveal changes in the relationship between the time-dependent attributes. The visualization of time-series plots in the context of the parallel-coordinates plot facilitates the exploration of time-related aspects of the data without the need to switch to a separate display. We provide a consistent set of tools for selecting and contrasting subsets of the data, which are important for various application domains.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Parallel-coordinates plots (PCP) are an established tool for the analysis of multivariate data sets. However, when it comes to the analysis of time-varying multivariate data, the traditional two-dimensional plot is of limited use. Time itself can be represented as an axis of the plot resulting in substantial overplotting already for small data sets [Ins01]. For very short discrete time series and few time-dependent parameters, the discrete points in time for each parameter can become individual axes, but this does not scale to longer time sequences or many parameters. Therefore, for the analysis of time-series data, a coordinated multi-view (CMV) approach is often adopted to display time-dependent variables in separate views and link them to a parallel-coordinates plot that only shows time-independent attributes (e.g. [KLM*12]) or a discrete time step of the time-varying attributes.

In contrast we developed a focus-and-context visualization, which integrates pseudo-perspective time-series plots between adjacent axes of a parallel-coordinates display (Figure 1). The underlying concept of our approach is that time can be considered as an extension in depth of the parallel-coordinates display. This concept was already suggested by Wegenkittl et al. [WLG97] for creating a surface rendering of the extruded parallel-coordinates plot. Barlow and Stuart [BS04] as well as Theron [The06] build on the general idea and provided the user with the basic possibility of navigating through a set of stacked parallel-coordinates plots–one plot per time step. However, this makes the detection of patterns in the time-series data of a single attribute quite difficult. In our visualization, we embedded a pseudo three-dimensional perspective of the time-series plots into the parallel-coordinates display to emphasize their spatial relationship. An independent parallel-coordinates panel that connects the two time-series plots can move forward and backward to explore their changes in relationship over time. We also provide brushing and selection tools to define subsets of the data records which work consistently across the parallel-coordinates displays and time-series plots.

Our work is motivated by our collaboration with a company that develops tools for robust design optimization of parametric simulations driven by e.g. a finite-element analysis. Their simulations of various kinds produce multivariate time-varying data sets. All of these data sets have a similar structure consisting of a set of independent input parameters, dependent responses and pre- and post-evaluated criteria. A particular set of values for the input parameters configures a simulation run which produces a set of mostly time-dependent responses such as displacement- or temperature-curves at various locations in an engine or crash pulses in a vehicle body (Figure 2). The design space exploration produces families of curves for each time-varying response, as also described in [KLM*12]. The trends of these time series and the interdependency of the parameters in time are a core subject of time-varying multivariate data analysis.

To our knowledge, our visualization is the first that relates two time-series plots to each other by a parallel-coordinates approach. Each time-series plot relates to a single attribute but shows the progression over time for many data records–in our use case simulation runs. One curve in one plot relates to exactly one curve in the other plot, both belonging to the same simulation run. Our time-

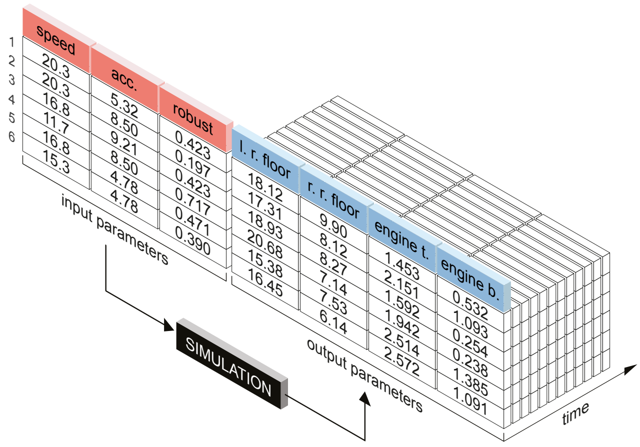

Figure 2: *The temporal and multivariate data cube consists of independent design parameters (red columns) and simulated responses (blue columns). Each simulation run (one row/slice) consists of a set of input values and the generated time-dependent output values.*

series vanishing-point widget uses a parallel-coordinates approach to show this 1:1 relationship for a single point in time and by moving it across time reveals changes in that relationship over time. Further important features of our widget are:

- Seamlessly integrates time-series plots in context of a parallel-coordinates display.
- Relies on an intuitive combination of the parallel-coordinates domain to the temporal domain.
- Allows an intuitive comparison of different points in time within the parallel-coordinates display.

We integrated this new widget in a prototypical visual analysis tool for multi-run time-dependent multivariate simulation data and explored its use for a real-world data set with expert users. Furthermore, we applied our tool to other domains where changes in the relationship of attributes over time play an important role in a problem analysis.

2. Related Work

Inselberg [ID90] invented parallel coordinates; the earliest ideas date back to the 1960s. A two-dimensional mapping of the multi-dimensional space is introduced by drawing each dimension as a vertical axis on a regular two-dimensional canvas. Each point in the multi-dimensional space is depicted as a sequence of line segments connecting the different vertical axes. In recent years, many important improvements with respect to interaction, drawing and data organization of parallel coordinates have been made. Inselberg's book [Ins09] and surveys on the state of the art [Mou11, HW13] provide an excellent overview. Furthermore, the book by Aigner et al. [AMST11] on visualization of time-oriented data is a comprehensive survey of the state of the art in this field. We focus only on publications that relate closely to our work on combining time series and parallel coordinates.

There are four basic approaches to consider time in the context of parallel-coordinates displays: (1) Time is simply included as an additional axis in the parallel-coordinates plot, e.g. Wong et al. [WB97]. Tominski et al. [TAS04] developed the TimeWheel, which is similar but does not display parallel axes. Instead the time axis is in the center of the display and the other axes are circularly arranged around it. In these approaches, the data record for each time step becomes a separate entry in the plot, which very quickly results in serious overplotting. (2) The attribute values for discrete time steps can be integrated as separate axes into the parallel-coordinates plot, which essentially turns it into a partial time-series plot. However, this is very limited with respect to the number of time steps and especially the number of separate time-dependent attributes. (3) The parallel-coordinates plot shows only a single time step, e.g. Barlow and Stuart [BS04] and Theron [The06]. Navigation in time is either provided by a time slider or through animation. (4) Blass et al. [BBP08] suggest using a coordinated multi-view approach to extract a single time-dependent axis and show a regular time-series plot for this axis. They argued that the use of a time slider by itself can be tedious and it is difficult to find a particular event. We support this approach, but also support a coordinated small-multiples view of time-series plots in addition to our embedded time-series view.

Wegenkittl et al. [WLG97] suggest a three-dimensional extension of parallel coordinates for displaying multivariate time-varying data sets. They create a parallel-coordinates system for every distinct time step and move them along the third spatial axis, which is similar to our mental model. However, they create a set of heightfields by connecting the line segments of the individual data items. Multiple heightfields are rendered transparently to reduce occlusion. Tominski et al. [TAS05] also suggest stacking single plots along the time axis, but effectively using a star plot arrangement of the parallel-coordinates axes. A tube-shaped surface is created by connecting subsequent time steps with quadrilaterals instead of a heightfield as in Wegenkittl et al. [WLG97]. However, both approaches scale only to very few simulation runs due to occlusion and the general visual complexity of the multiple potentially interpenetrating surfaces. Furthermore, three-dimensional view point navigation is necessary to view and explore the model.

Johansson et al. [JLC07] also investigated temporal parallel coordinates. They create quadrilaterals between two adjacent axes, which are defined by the data values for consecutive points in time. The set of quadrilaterals is rendered with a fixed density such that time is considered, e.g. earlier data samples appear in the foreground and later ones in the background or vice versa. Accumulating densities then separates regions of high variation from regions of low variation. In essence, they implicitly also use the idea of Wegenkittl et al. [WLG97] by extruding parallel-coordinate plots in depth and connecting them by quadrilaterals. However, they provide only an orthogonal view and a parallel projection along the depth axis instead of free view point navigation and focus on visualizing changes over time by adding depth cues and temporal density. Furthermore the relation between dimensions resulting from the data behavior along a time step is not properly reflected by the fixed-density model. This is a result of the point-line duality between Cartesian and parallel coordinates and its strong effect on density. Therefore Grottel et al. [GHWG14] use a physically motivated density and mass conversation to create continuous-time parallel coordinates which enhance the depth cues and temporal density. These approaches reveal general trends and are thus limited in the detail of temporal changes they can provide. Therefore, they are unlikely to scale to simulation problems consisting of many simulation runs, where each run produces a time-dependent multi-attribute data set.

3. The Time-Series Vanishing-Point Plot

As the fundamental model of our approach, we propose connecting the two-dimensional parallel-coordinates domain to the two-dimensional spatiotemporal domain of the time-series plots by considering time as the third dimension of the parallel-coordinates plot. This construction creates a three-dimensional object where the parallel-coordinate plots are parallel to the x-y plane and time is arranged in depth as the z-axis. A cross section in time is a regular parallel-coordinates plot for that particular time step. An x-z cross section through one of the parallel-coordinates axes results in a time-series plot. Looking at it from the top (down the y-axis), one could imagine a set of interwoven heightfields. Each heightfield is created by connecting the parallel-coordinate lines of a single simulation run across time (Wegenkittl et al. [WLG97]).

However, the visualization itself has now become a complex object that lives in three-dimensional space. Tools are needed to explore the space and navigate through the visualization. Unconstrained spatial navigation could allow the user to fly through the visualization to define arbitrary views. However, it would be difficult to make sense of the observed data due to occlusion and perspective. Instead, we propose a more gentle approach that seamlessly integrates into regular parallel-coordinates displays and requires little navigation.

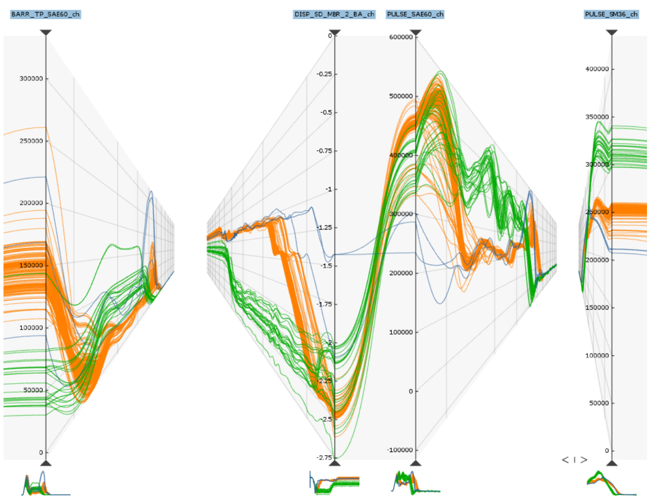

Figure 3: *Multiple vanishing-point widgets for adjacent parameters in the same parallel-coordinates plot. The vanishing point of the right widget was moved to the right to focus on the left time-series plot.*

3.1. Visual Design

We developed the time-series vanishing-point widget to exploit the intuitive mental model of the underlying three-dimensional visualization object. This widget embeds time-series plots between two adjacent axes of the parallel-coordinates display using a pseudo-perspective rendering based on a single vanishing point (Figure 1). The widget is only activated on demand and can be used between multiple axes pairs at the same time. However, we do not tie all the opened vanishing-point widgets to a single vanishing point and effectively create a multi-perspective display (Figure 3). While this might seem counterintuitive at first, it is of great advantage to be able to independently use multiple vanishing-point widgets. Otherwise, one or more of the multiple time-series plots would be extremely distorted or even be occluded if the widget was located farther left or right of a single central vanishing point. Furthermore, parts of the time-series plots which are in front of the global time step of the reference plot are clipped to avoid occlusion.

While showing just the time-series plots in context of the parallel-coordinates plot is already useful, it partially breaks with the parallel-coordinates paradigm and creates two disconnected plots. The visual analysis of the relationship between the two adjacent time-series plots is no longer easily possible. However, one curve in the left plot relates to exactly one curve in the right plot, both belonging to the same data record. Thus, as a central element of our widget, we integrated an independent parallel-coordinates panel to show this 1:1 relationship for a single point in time (Figure 1).

3.2. Interaction Design

Our reference display is the parallel-coordinates plot showing a number of time-independent parameters, as well as a number of time-dependent parameters. For the latter, we support regular navigation through time using various means such as a time slider, also suggested by e.g. [BS04, The06, KLM*12]. We refer to this time step of the reference display as the global time step.

The third dimension of the parallel-coordinates plot becomes only apparent once you start to use the time-series vanishing-point widget. The core functionality of the widget's parallel-coordinates panel is that it can move forwards and backwards to reveal the evolution of the relationship between the left and right set of curves over time.

In our system it is possible to insert the same axis more than once. This feature allows direct comparisons of the same pair of axes at different points in time. The left panel in Figure 4 shows a point in time prior to the right panel. The relationship between the two clusters is inverted over time. This behavior becomes immediately apparent when dragging the panel back and forth, which would also reveal when the inversion happens. However, for certain scenarios, it is more convenient to duplicate a vanishing-point plot by inserting one or more axes twice (see Figure 4). Color-coded clusters allow to estimate changes in relationship even without the translucent panel (Figure 3). However, only the integrated panel enables a direct exploration of changes in the relationship between the curves on both sides and, thus, to discover such connected clusters in the first place.

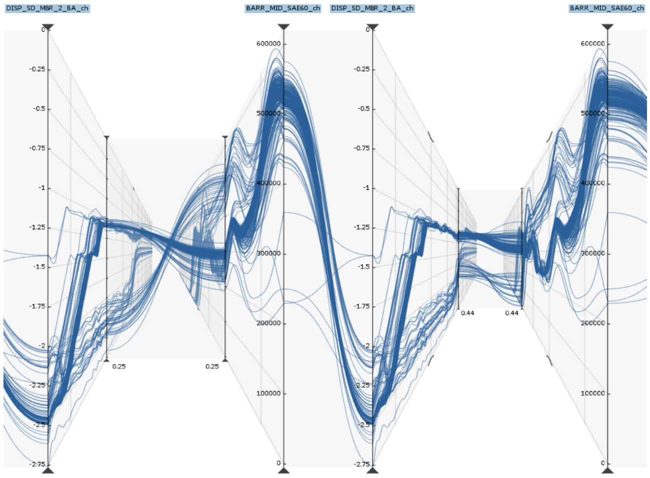

Figure 4: *Two vanishing-point widgets which show the same time-series plots. The relationship between the two clusters in the two times-series plots gets inverted between the two time steps.*

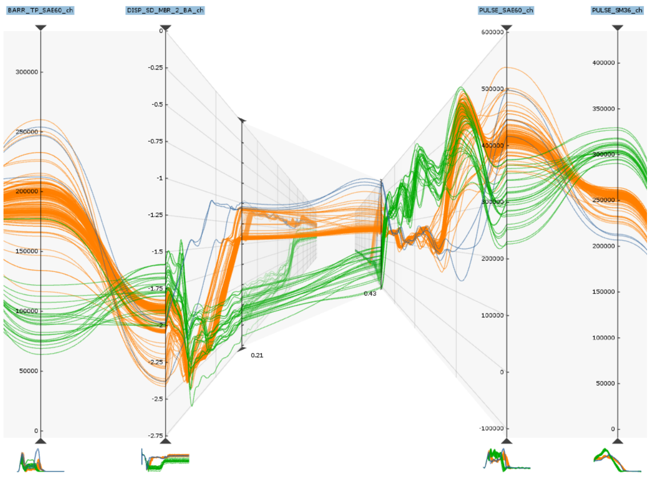

Figure 5: *The parallel-coordinates panel connects two different time steps of the two adjacent time-series plots to reveal delays between the time-dependent attributes. The orange and green clusters on the left side seem to form at about the same point in time but the relationship between the axes gets inverted over time.*

Related time series are often slightly delayed against each other since they might be affected by the same phenomenon, just at different points in time, e.g. the temperature increase along a bar that is heated up from one end. To reveal such delays between two time-series plots, the parallel-coordinates panel of the widget can be tilted around the vertical axis by separately setting the local time step of the left and right time-series plot (Figure 5). This way it shows the parallel coordinates between different points in time of the adjacent attributes. If they are related and there is approximately a constant delay between the two time series, moving the panel forward and backward would always show a familiar parallel-coordinates pattern of e.g. a linear relationship between the attributes. If the delay is non-constant, one would see a change of the parallel-coordinates pattern over time.

When adjacent vanishing-point widgets are opened, we have two ways to deal with this situation: Either we simply show two completely independent vanishing-point widgets next to each other or we merge them (Figure 6 top). The merged view shows time-series plots only for the leftmost and rightmost axes to avoid occlusion. The local time step on each axis can now be shifted relative to the other axes to create a particular pattern in time (Figure 6 bottom). The entire fence can then be moved forward and backward in time to explore changes of the relationships of multiple attributes.

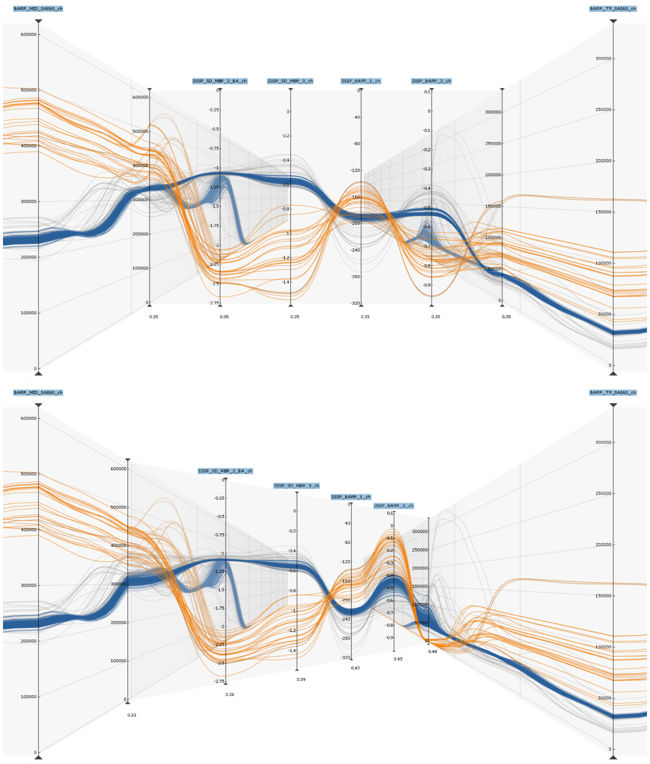

Figure 6: *Multiple adjacent vanishing-point widgets are opened and shown in a merged view. The axes of the vanishing-point widget display the same local time step (top), but can also be configured to show an individual time step at each axis (bottom).*

3.3. Design Discussion

Time-series plots visually aggregate the individual lines and thus might reveal structures such as clusters or bundles if hundreds of lines are drawn into the diagram. Our vanishing-point widget allows the user to explore the temporal relationships between such structures of two adjacent time-series plots. We favor a slight perspective projection over a parallel one for the visualization of the embedded time-series plots for several reasons. First, it emphasizes the intuitive mental model of the data. Orthographic projections are well suited for actual measurements but may result in incorrect estimations of sizes. In our opinion, the only disadvantage is that fewer pixels are available for the height of the time series towards the vanishing point. However, we use a grid to facilitate the estimation and comparison of values in a time series. In addition, the perspective drawing of the line charts serves as a natural focus-and-context display that provides most detail around the current time step and visually aggregates the information further in time.

The scalability of our approach is limited in many respects. The number of simultaneously opened vanishing-point plots is limited, since our widget needs more space to be effective in comparison to the regular spacing between two adjacent axes of a parallel-coordinates plot. Often only one plot is used at a time. In our experience, the length of the time series is also limited to a few thousand time steps. Longer time series require aggregation techniques and potentially non-linear time scales. Our examples contain up to 400 curves in each time-series plot.

Konya et al. [KLM*12], argued it is not sufficient to explore trends in time of individual parameters. As a solution, they display a single time-series plot next to the parallel-coordinates display. While this helps, it requires the user to focus back and forth between the two displays while our approach embeds the time-series plots within the parallel-coordinates plot and allows the direct exploration of the relationship of the time series.

If only the correlation between parameters in a particular dataset is relevant, it would be also possible to embed a correlation graph between two axes. However, due to the complex physical models that are underlying the simulation runs, linear correlations rarely exist. It is also possible to explore the relationships between two parameters at the same or different time steps via a scatter plot that could be embedded between the two axes, either with or without the time-series plots on the left and right side. However, the relationship of a point in the scatter plot to the corresponding curve on the left and right axis is hidden and needs to be explicitly revealed by techniques such as brushing and linking or color-coding. Alternatively temporal glyphs in a small-multiples layout could be embedded between two axes of the parallel-coordinates plot [FFM*13]. However, trends, similarities and differences across time of subsets of simulation runs and interdependencies across parameters would be very difficult to explore.

4. Coordinated Multi-View Prototype

The parallel-coordinates plot with integrated time-series plots is the major and dominant display of our visual analysis prototype (Figure 7). It supports a variety of advanced interactions beyond the vanishing-point widget to enable the analysis of real-world data sets such as arbitrarily adding, removing and reordering the individual axes or globally adjusting the translucency similar to [AdOL04, NH06, DHNB09, PBM05]. For reducing overplotting, we support extended areas [ROF12] which spread each datapoint along the axis according to the distribution of values along that axis. This approach also simplifies brushing interactions in these situations. Horizontal zooming increases the space between all axes and panning slides the entire plot left or right.

Parallel-coordinates plots horizontally extend beyond the screen for a large number of parameters. In order to provide an efficient navigation and aid the user in keeping track of their current position in the information space, we have integrated an overview into the

Figure 7: *A snapshot of our application showing a crash simulation data set. (a) The constraint set widget, containing the two predefined and one user-defined data subsets. (b) The parallel-coordinates canvas displays a subset of the attributes, whereas the activated overview shows them all. (c) The parameter repository displays mainly time-dependent parameter. (d) Multiple canvases showing different kinds of plots.*

parallel-coordinates display showing the complete plot while neglecting details (Figure 7b). Changes in the parallel-coordinate plot are immediately reflected by the overview that also enables fast axis rearrangement by directly dragging axes from the overview into the actual plot and vice versa.

The time-extended parallel-coordinates display can be accompanied by various kinds of plots including separate time-series plots, scatter plots and scatter-plot matrices. Intelligent drag-and-drop functionality allows to open new canvases with plots of one or several selected parameters defined by their respective type. A single parameter creates a histogram, a single time-dependent one results in a time-series plot, whereas a scatter plot will be generated by dragging two parameters onto the new canvas and a scatter-plot matrix, when using more than two parameters. The canvases are freely arrangeable and can be nested arbitrarily deep.

The views are coordinated through (1) the global time step, (2) local time steps of the currently opened vanishing-point plots and, (3) subsets of the simulation runs defined by a set of selection and brushing tools. Global and local time steps can be manipulated via the global time slider, the vanishing-point widget or the time-series plots and always remain consistent. Furthermore, hovering over an axis highlights all axes in the other views showing the same attribute. Multiple subsets can be defined either by adjusting ranges with sliders and invoking specific values with extended areas or by directly brushing paths in-between the axes or within the vanishing-point plot. The first method intersects across all axes and defines the two disjoint sets of selected and unselected data records. The two subsets and the operations are permanent and can be adjusted later on by the appropriate constraint representation. In contrast to the first one and its corresponding subsets the brushing operations and resulting subsets are transient, since the brushing rectangle is only visible during the operation and can not be adjusted afterwards. Also the subsets to which data records are either added or removed by a brushing operation can be created or removed on demand (Figure 8). Each defined data subset is represented in the list of constraint sets (Figure 7a). The displayed widget enables users not only to create and remove the transient brush sets, but also to adjust the color and opacity. Subsets can be temporarily hidden to reduce visual clutter. Selecting an item in the list, causes the data records of the corresponding subset to be drawn on top of all other data items. This way the focused subset is not occluded and therefore allows a better exploration. Changes in the appearance and ordering of data subsets are immediately reflected in all views.

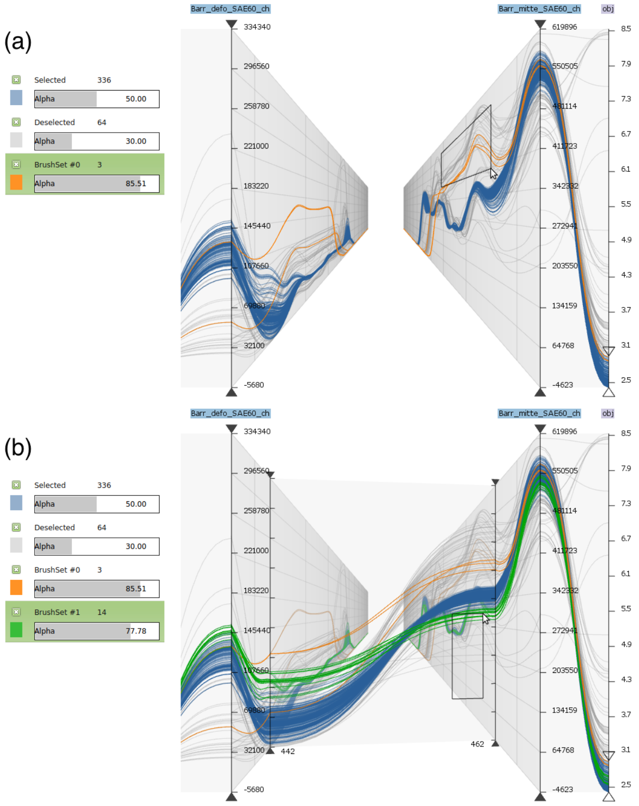

Figure 8: *Selection and brushing in action. (a) A subset of the data records has been selected by the range sliders of the rightmost axis and thus showed in blue. Some outliers are being brushed in the right time-series plot, turned orange, and appeared as new subset in the list on the left. (b) By brushing on the parallel-coordinates panel of the vanishing-point plot, another subset is defined in green.*

5. Implementation

Our application has been implemented using C++, OpenGL and the OpenGL Shading Language (GLSL). GUI elements and the OpenGL context are provided by Qt, whereas all content itself is implemented using different stages of the graphics pipeline. The different plots like scatter plots or the subplots (between two axes) of the parallel-coordinates plot are organised as scene graph and composed of vertex buffer objects (VBO), index buffer objects (IBO), their corresponding shader programs and necessary uniforms. Evaluating Bezier curves on the CPU would result in low performance so we used two shader stages; Tessellation Control Shader (TCS) and Tessellation Evaluation Shader (TES) introduced in OpenGL 4.0. We define the number of line segments for the Bezier curve within the TCS (outer tessellation levels) and evaluate the Bezier curve at each particular vertex location within the TES. The control points for the Bezier curve are sent down the pipeline as a patch primitive consisting of four vertices. The final output of the tessellation stages is a set of primitives, in our case, a line strip. Transparency is provided by utilizing the hierarchical structure of the rendering scene graph for drawing the objects back-to-front and blend them correctly when displayed on top of each other. For axis labeling we used the extended wilkinson algorithm [TLH10].

The current implementation was tested with a data set consisting of 400 simulations runs, 58 attributes, and more than 2400 time steps. Data handling, rendering and interaction could be smoothly performed even on 4k screen resolutions. The performance fulfilled the requirements by our collaboration partner. Future data sets will grow beyond 1000 simulation runs, 100 attributes and 10 000 time steps. The attribute repository and the focus on a subset of the parameters are always an option to improve performance. Scaling towards more time steps requires the use of temporal aggregation techniques.

6. Expert Feedback and Discussion

Our system was reviewed with experts from different domains using their respective time-varying data sets. Our industrial partner, for example, focused on the challenge of optimizing a vehicle body for best safety of the passengers. The goal is to reach a mass reduction of a component at the best possible stability (moderate displacements, smooth pulses) of the body during a crash. The experts helped us during development to proceed with their knowledge in engineering, suggested new ideas and provided us with sample data. At several occasions during the development we went through case scenarios for revealing the problems and caveats of the current software prototype.

Results of such a crash simulation could be the variation in time of energies and displacements at different positions of the automobile body or the acceleration of the limbs of a passenger. The goal is to find designs that do not compromise the passengers while assuming moderate and manageable stiffnesses of the involved components like side- and cross-members, bumper and the surrounding body structure. The resulting multivariate data set contains both the design parameters and the simulated responses for each simulation run. Eventually, their data consisted of 400 simulation runs over 58 attributes (anonymized for confidentiality) consisting of 6 scalar input parameters, 33 scalar responses and criteria and 19 time-dependent outputs. Each time-dependent attribute of a simulation run spans over 2401 time steps.

The core task of the experts is to find the good value combinations for the input parameters. For this task, they defined an objective that considered displacements at various components and the matching of pulse curves to their reference counterparts. Although, as stated by our experts "*The original objective was rather complex and rough, the embedded vanishing-point plots clearly expressed the potential that could lead to a couple of pleasing designs. Furthermore the interactive visualization gave us an indication that the investigation of different – here second grade successful – areas of the design space could be of some value as other clusters with good properties (i.e. response combinations) could be easily identified.*" The identification of these clusters of simulation runs was performed by brushing on the time-series plots and on the parallel-coordinates panel.

The vanishing-point widget evolved from a vague suggestion expressed by one of the company's analysts. With their usual tools the simulation analysis was more difficult since either time-series plots or parallel coordinates for a single time step could be shown. The vanishing-point widget was considered to be particularly effective identifying clusters of simulation runs with good properties across multiple attributes. They stated that the integrated view better supports their mental model, because navigating in time is now effortless and more direct. With our prototype it became easier to focus on the analysis itself which confirms that that our system is a major improvement in usability compared to their existing analysis tools. They also found the interactive small-multiples display of the time-varying responses very useful and suggested to use it as a storage space for important parameters. Thanks to our optimized GPU implementation, our system always performed at a frame rate of 60 Hz which allowed a fluid interaction with the visual analysis tool which was also considered important. Overall, our approach enables the analysts to formulate, verify or reject hypotheses far more conveniently than before. Particularly, in the given real-world data set "*the objective had identified some designs that had partially good properties where the matching of the pulse curves was acceptable but individual short, yet extreme, displacements of these designs during the virtual crash, immediately visible in the vanishing-point plot, entails enormous potential for passengers injuries and thus have to be discarded.*"

We reviewed our system with a second expert group from the civil engineering department of our university. They oversee the modeling of the restoration of an artificial dam. In order to find a dam's weakest spots, various models are evaluated by simulation runs to investigate the restored dam's behavior while the reservoir is being filled. Input parameters are permeability for temperature changes and liquids, material density, fill states and others. Simulation results include strain, stress and displacement at particular sample points on the surface and inside the dam. A coarse 3D dam model was rendered using the GiD [Int] post-processor which maps values to colors for visualizing a particular attribute and time step. For an in-depth analysis, Matlab was used to plot aggregated and derived attributes over time.

Contrary to our expert review above, where each line in the parallel-coordinates plot was a simulation run, here each line represents a sample point of the dam. One crucial result is the drift angle of the dam at the sample points. Dams are often constructed to lean towards the reservoir while the reservoir is empty. The further the water level rises, the more the dam leans in the other direction (away from the reservoir). The point in time when the dam starts to lean outwards varies depending on the dam's parameters. During this process, the displacements for all other directions must remain stable for a successful outcome (see Figure 9).

As stated by our experts our vanishing-point plot is the first visualization where they are able to recognize at a single glance whether the dam behaves as expected, i.e. if the pattern changes from a straight to an inverse relationship between the displacements in y- and z-direction is clearly visible (see Figure 9). Any deviation from that behavior is a result of a failed simulation either due to an incorrect choice of parameter values, an incorrect structural model or simply an error of the numerical solver. Hence, our approach serves as a very convenient simulation verification method, which is the very first core task of our experts. To our knowledge it is hard to come similarly fast to this conclusion by any other visualization. For example, a scatter plot is also capable of showing the relationship between two attributes for a single time step, but when it is moved through time, it completely relies on visual memory and does not provide any context for orientation. A dam model depicted in 3D also relies solely on visual memory if the displacement over time is shown as an animated color mapping. Showing the actual displacements in 3D would be barely visible. In contrast, with our vanishing-point widget, the context of the time-series plots provides orientation and strengthens the visual memory of the moving parallel-coordinates panel.

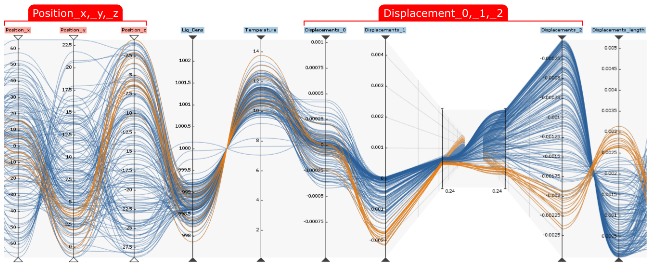

Figure 9: *The vanishing-point plot shows an ideal dam displacement over time (see Displacement_0,_1,_2). The displacement attribute expresses the shift of the individual points in their respective direction ($0 \to x$, $1 \to y$, $2 \to z$). The sample points with the largest displacements at the end of the fill process are shown in orange. They are located in the upper center of the dam (see Position_x,_y,_z – x along the dam, $x = 0$ at the dam's horizontal center, $z = 0$ at the top of the dam , $+y$ in direction towards the reservoir).*

7. Conclusion and Future Work

We present a novel approach for exploring time-dependent parameters in a parallel-coordinates display which relies on the intuitive mental model that time can be considered as the third dimension of a regular parallel-coordinates plot. Our vanishing-point widget can be seamlessly integrated between two or more axes to avoid a context switch to another window or view. Only when it is activated it does reveal the temporal progression of the involved attributes and their relationship. Moving the parallel-coordinates panel backwards and forwards between the two adjacent time-series plots reveals changes in their relationship over time. The time-series plots on the left and right side provide a reference for the changes seen in the parallel coordinates and aid the visual memory. Constant latencies and delays between time-dependent signals can be detected and analyzed very easily by tilting the parallel-coordinates panel appropriately. Our parallel-coordinates view is enhanced by an overview at the top, which provides convenient orientation and navigation for data sets consisting of many attributes. The overview can also be used to rearrange axes.

We integrated our extended parallel-coordinates plot into a coordinated multi-view visualization system, which provides an attribute repository and a fully configurable small-multiples view for time-series plots and scatter plots. The small-multiples displays often serves as a storage space for important parameters that have been identified throughout the analysis process. For the exploration and contrasting of subsets of the data records, we provide a powerful set of selection and brushing tools which work consistently across all of the different visualizations.

Future work will concentrate on managing larger data sets with respect to the number of records, the number of parameters and the length of time series. We will also link to a visualization of the underlying simulated 3D model in a similar way as e.g. [BBP08]. Furthermore, our simulation experts suggested that we should support the comparison of multiple snapshots in time of sets of attributes. To this end, we will extend our small-multiples view to integrate multiple parallel-coordinates displays, which would only show a subset of the axes and a particular time step each. We already support one way to address this problem by simply providing the option of having the same attribute more than once in the parallel-coordinates plot. Hence, by opening multiple instances of our vanishing-point plot, we enable the direct comparison of different points in time without leaving the parallel-coordinates paradigm. Furthermore, we plan to perform a user study to investigate the performance and perceptional aspects of our integrated vanishing-point widget in comparison to a classic coordinated multi-view display.

Overall, the coordinated multi-view visualization system was highly appreciated by our experts for its configurability, its versatility in analysis, its fluent rendering and its intuitive interactive capabilities. The discussions and reviews clearly confirmed the value of our times-series vanishing-point widget: it provides insight into the data that was previously hard to come by with any other kind of visualization. The experts suggested that our widget should be available as a standard tool in parallel-coordinates displays for time-dependent data. Thus, our focus-and-context technique for seamlessly integrating time-series plots into parallel-coordinates displays contributes to the fundamental set of tools and techniques for the analysis of multivariate time-varying data.

Acknowledgements

The authors thank the entire team of Dynardo Software & Engineering GmbH as well as Prof. Dr. Tom Lahmer and Dr. Long Nguyen Tuan from the Institute of Structural Mechanics at Bauhaus-Universität Weimar for their support with data sets and expertise.

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) under grant 03IP704 (project Intelligentes Lernen) and grant 03IPT704X (project Big Data Analytics).

References

[AdOL04] ARTERO A. O., DE OLIVEIRA M. C. F., LEVKOWITZ H.: Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of the IEEE Symposium on Information Visualization* (Washington, DC, USA, 2004), INFOVIS '04, IEEE Computer Society, pp. 81–88. 5

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*, 1st ed. Human-Computer Interaction. Springer Verlag, 2011. 2

[BBP08] BLAAS J., BOTHA C., POST F.: Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov. 2008), 1436–1451. 3, 9

[BS04] BARLOW N., STUART L.: Animator: a tool for the animation of parallel coordinates. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on* (July 2004), pp. 725–730. 2, 3, 4

[DHNB09] DIETZSCH J., HEINRICH J., NIESELT K., BARTZ D.: SpRay: A visual analytics approach for gene expression data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* (Oct 2009), pp. 179–186. 5

[FFM*13] FUCHS J., FISCHER F., MANSMANN F., BERTINI E., ISENBERG P.: Evaluation of alternative glyph designs for time series data in a small multiple setting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 3237–3246. 5

[GHWG14] GROTTEL S., HEINRICH J., WEISKOPF D., GUMHOLD S.: Visual analysis of trajectories in multi-dimensional state spaces. *Computer Graphics Forum 33*, 6 (2014), 310–321. 3

[HW13] HEINRICH J., WEISKOPF D.: State of the art of parallel coordinates. In *STAR Proceedings of Eurographics 2013* (2013), Association E., (Ed.), pp. 95–116. 2

[ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st Conference on Visualization '90* (Los Alamitos, CA, USA, 1990), VIS '90, IEEE Computer Society Press, pp. 361–378. 2

[Ins01] INSELBERG A.: Conflict detection and planar resolution for air traffic control. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE* (2001), pp. 1200–1205. 2

[Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2009. 2

[Int] INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING (CIMNE): GiD: Pre and Post Processor. `http://www.gidhome.com/`. [Online; last accessed 11-March-2016]. 8

[JLC07] JOHANSSON J., LJUNG P., COOPER M.: Depth cues and density in temporal parallel coordinates. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC Conference on Visualization* (Aire-la-Ville, Switzerland, Switzerland, 2007), EUROVIS'07, Eurographics Association, pp. 35–42. 3

[KLM*12] KONYHA Z., LEŽ A., MATKOVIĆ K., JELOVIĆ M., HAUSER H.: Interactive visual analysis of families of curves using data aggregation and derivation. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies* (New York, NY, USA, 2012), i-KNOW '12, ACM, pp. 24:1–24:8. 2, 4, 5

[Mou11] MOUSTAFA R. E.: Parallel coordinate and parallel coordinate density plots. *Wiley Interdisciplinary Reviews: Computational Statistics 3*, 2 (2011), 134–148. 2

[NH06] NOVOTNY M., HAUSER H.: Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Sept. 2006), 893–900. 5

[PBM05] PRADHAN K., BARTZ D., MUELLER K.: *SignatureSpace: a multidimensional, exploratory approach for the analysis of volume data*. Tech. rep., Universitätsbibliothek Tübingen, Wilhelmstr. 32, 72074 Tübingen, 2005. 5

[ROF12] RIEHMANN P., OPOLKA J., FROEHLICH B.: The product explorer: Decision making with ease. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2012), AVI '12, ACM, pp. 423–432. 5

[TAS04] TOMINSKI C., ABELLO J., SCHUMANN H.: Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM Symposium on Applied Computing* (New York, NY, USA, 2004), SAC '04, ACM, pp. 1242–1247. 3

[TAS05] TOMINSKI C., ABELLO J., SCHUMANN H.: Interactive poster: 3d axes-based visualizations for time series data. In *IN POSTER PROCEEDINGS COMPENDIUM, IEEE SYMPOSIUM ON INFORMATION VISUALIZATION (2005* (2005), pp. 49–50. 3

[The06] THERON R.: Visual analytics of paleoceanographic conditions. *2012 IEEE Conference on Visual Analytics Science and Technology (VAST) 0* (2006), 19–26. 2, 3, 4

[TLH10] TALBOT J., LIN S., HANRAHAN P.: An extension of wilkinsonâĂŹs algorithm for positioning tick labels on axes. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2010). 7

[WB97] WONG P. C., BERGERON R. D.: 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques* (Washington, DC, USA, 1997), IEEE Computer Society, pp. 3–33. 3

[WLG97] WEGENKITTL R., LÖFFELMANN H., GRÖLLER E.: Visualizing the behavior of higher dimensional dynamical systems. In *Proceedings of the 8th Conference on Visualization '97* (Los Alamitos, CA, USA, 1997), VIS '97, IEEE Computer Society Press, pp. 119–ff. 2, 3