

Interactive Visual Analysis of Families of Curves using Data Aggregation and Derivation

Zoltán Konyha
VRVis Research Center
Vienna, Austria
Konyha@VRVis.at

Alan Lež
VRVis Research Center
Vienna, Austria
Lez@VRVis.at

Krešimir Matković
VRVis Research Center
Vienna, Austria
Matkovic@VRVis.at

Mario Jelović
AVL-AST d.o.o.
Zagreb, Croatia
mario.jelovic@avl.com

Helwig Hauser
University of Bergen
Bergen, Norway
Helwig.Hauser@UiB.no

ABSTRACT

Time-series data are regularly collected and analyzed in a wide range of domains. Multiple simulation runs or multiple measurements of the same physical quantity result in ensembles of curves which we call *families of curves*. The analysis of time-series data is extensively studied in mathematics, statistics, and visualization; but less research is focused on the analysis of families of curves. Interactive visual analysis in combination with a complex data model, which supports families of curves in addition to scalar parameters, represents a premium methodology for such an analysis. In this paper we describe the three levels of complexity of interactive visual analysis we identified during several case studies. The first two levels represent the current state of the art. The newly introduced third level makes extracting deeply hidden implicit information from complex data sets possible by adding data derivation and advanced interaction. We seamlessly integrate data derivation and advanced interaction into the visual exploration to facilitate an in-depth interactive visual analysis of families of curves. We illustrate the proposed approach with typical analysis patterns identified in two case studies from automotive industry.

Categories and Subject Descriptors

I.3.6 [Computer Graphics]: Methodology and Techniques—*Interaction techniques*; I.6.6 [Simulation and Modeling]: Simulation Output Analysis

General Terms

Design

Keywords

Interactive visual analysis, knowledge generations, families of curves, attribute derivation.

1. INTRODUCTION

Generating useful knowledge from the information that is often only implicitly available in complex data sets is one of the key challenges in analysis. Interactive visual analysis (IVA) has proven itself as a valuable method of getting insight into, understanding, and analyzing complex data. However, IVA is still not well integrated into the whole analysis work flow. There is a multitude of powerful methods presented, but, unfortunately, they often remain isolated.

Our work is motivated by case studies we have done with domain experts from different fields including engineering [13, 14, 16] and medicine [15]. We discovered that many details of the data sets from those very different problem domains can be represented as *families of curves* [14]. We could also identify similar procedures in the analysis of families of curves. We suggest that representing curves as an atomic type opens new analysis possibilities. We focus on a set of tools for the analysis of data that contains families of curves.

The analysis of curves is a well known and extensively researched topic in science and mathematics. However, there is not much research on the analysis of entire families of curves. IVA offers an effective and efficient opportunity to analyze even larger families of curves [14]. The state of the art follows the visual information seeking mantra [21]: *overview first, zoom and filter, then details-on-demand*. This approach has often been proven powerful, but especially when working with families of curves, there are cases when zooming and filtering is not sufficient. Our work is in part motivated by curve sketching, a familiar topic from high school. We seek to extend the data set by computing new attributes and additional derived curves when analyzing families of curves; just as we compute attributes (e.g. extrema) and curves (e.g. first derivative) in curve sketching. Curve sketching helps us in the analysis of single curves. In this paper, we apply related methods to entire families of curves.

Traditional visualization systems limit data manipulation to filtering and require more complex processing to be performed in a separate step *before* the visual analysis. When that pre-processing is designed, one needs to estimate what properties of the data are expected to be of interest. Such a priori knowledge is often not available, in particular not in more intricate analysis cases. In this paper we present

a set of tools which extends the conventional approach and facilitates on demand data generation. By allowing the synthetic extension of data by attribute derivation (in addition to filtering, of course) at any time, new analysis possibilities arise that are useful in particular for experts. An integrated system makes it possible to reveal deeply hidden information in the data without requiring detours or a priori knowledge to design the pre-processing.

The main contribution of this paper is not a new visualization method or a new view. It is a set of analysis procedures, cleverly combined in one toolbox, which facilitates the deep and flexible analysis of complex data containing families of curves. We argue that a rich set of tightly integrated (general) analysis mechanisms can help in a wide range of application scenarios. We classify the analysis process itself and suggest what should be included in an advanced IVA framework. We have identified three levels of complexity in the analysis process. The first two levels represent the current state of the art. We propose a third level that includes advanced interaction and on-demand attribute derivation in order to facilitate the exploration of deeply hidden details. We claim that there are two distinct ways to explore hidden features. One is to keep data intact and offer more complex interaction. The other possibility is to extend the data to be more complex and keep interaction simple. We demonstrate that the two approaches are complementary; there is no universally preferable way. We use examples from several case studies with domain experts from various fields to illustrate the possible use of the proposed techniques.

2. RELATED WORK

In this section we summarize work related to ours, including the integration of computational methods into IVA, the visualization of multivariate time-dependent data, and coordinated multiple views.

Interactive visual analysis is powerful, but cannot capture certain aspects of complex data sets in its standard form. Visual analytics research [12, 22] suggests that computational data analysis methodologies, such as statistics, data mining, or machine learning should be integrated with IVA to create a knowledge discovery framework. The survey by Bertini and Lalanne [5] exemplifies that computational and visual methodologies are complementary. It is promising to aim for solutions where interactive, visual approaches are tightly integrated with automated, computational ones, such that an efficient iterative approach to data analysis becomes possible. Such mixed-initiative knowledge discovery systems take the best of human and machine capabilities [5].

Reviewing the large body of literature on the visualization of *time-dependent data* is beyond the scope of this paper. Readers are directed to the recent book by Aigner et al. [1]. Visualization of *multivariate data* has also been researched for decades. In this paragraph we focus on methods that visualize a special type of multivariate time-dependent data, namely families of curves. Families of curves are commonly depicted in line charts [14] or dense pixel displays [18]. Spatio-temporal data often includes families of curves. In the book by Andrienko [2], visualization methods for spatio-temporal data and common analysis tasks, including computing differences and performing advanced queries, are discussed.

a.

Record #	Time step	Diff_H	Diff_V	Tropical temp.
1	1	1250	7.5e-5	24.50
2	2	1250	7.5e-5	24.50
3	3	1250	7.5e-5	24.50
...
500	500	1250	7.5e-5	24.30
501	1	1250	7.7e-5	24.20
...
1001	1	1250	7.9e-5	24.01
...
50000	100	2500	9.5e-5	24.25

b.

Record #	Diff_H	Diff_V	Tropical temp. (t)
1	1250	7.5e-5	
2	1250	7.7e-5	
3	1250	7.9e-5	
...
100	2500	9.5e-5	

A Family of Curves

Figure 1: Two approaches to the management of curve data. (a) Time is represented by adding an additional dimension. Records 1 to 500 represent one curve. (b) Curve is represented as an atomic type in the data. Columns of this table are families of curves.

Visual methods that support the exploration of complex systems are actively researched. The Influence Explorer [23] is one of the early examples. Here we focus on systems that make an effort to support the analysis of information that is not explicitly represented in the data set. The interactive derivation of new data attributes has been identified as one of the key aspects of visual analysis [8], because it leads to a useful feedback loop in the analysis process [12]. Still, it has not been widely implemented in existing systems. Cross-filtered views [24] support the derivation of new data attributes. Kehrer et al. [11] incorporated the computation of statistical aggregates into the analysis. Doleisch [6] proposed attribute derivation to allow interactive feature specification. Berger et al. [4] presented a system to enable the continuous analysis of a sampled parameter space using methods from statistical learning to predict results that are not available. However, similar approaches have not been incorporated into the analysis of families of curves yet.

Coordinated multiple views (CMV) combine different views on the same data in such a way that a user can correlate the different views. Subsets of data can be interactively selected and the selected subset is highlighted in all other views in a consistent manner. This enables users to effectively explore and analyze high dimensional data. The survey by Roberts [20] provides an overview on the state of the art in coordinated multiple views.

3. DATA MODEL AND EXAMPLE

Data in databases and also in IVA applications are generally stored as records that consist of attributes. This concept is well known, records can be considered as points in an n dimensional space where n is the number of attributes. All attributes (dimensions) are scalars, either numeric or categorical. We omit a more detailed discussion including nominal and ordinal types here. Time-dependent data can be represented by adding time as an additional dimension.

Based on our previous work [14], we describe an enhanced data model that natively supports time dependent dimensions. We use an example from climate research [3] to illustrate the data model. Climate researchers develop models that can predict future climate development. The models are often tested and validated against known past scenarios.

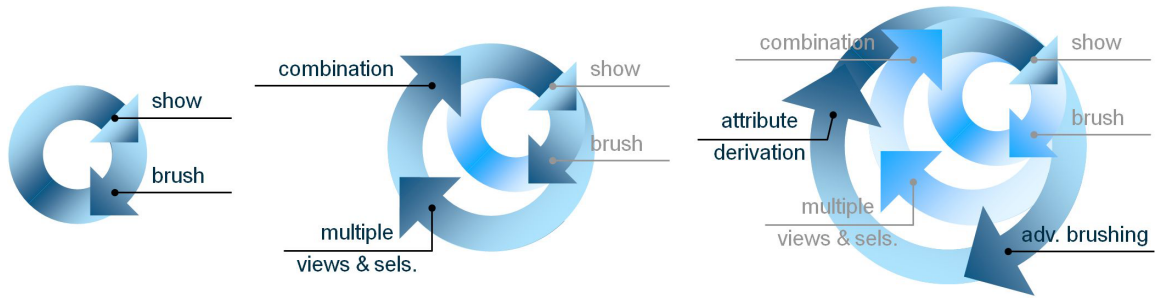


Figure 2: Interactive visual analysis on three levels of complexity. The first level is linking and brushing with one brush. On the second level, more views are used and brushes are combined with logical operators. On a third level, advanced interaction (brushing) and attribute derivation are added.

In our example, the simulated climate response to the outburst of meltwater from Lake Agassiz was studied [3]. The diffusivity parameters of the ocean model (two scalars, $diff_H$ and $diff_V$) were varied across simulation runs. There were 10 variations of each parameter, producing 100 individual simulation runs. In each simulation run, time series data is generated, including surface air temperature over land and oceans, global precipitation, etc. Each simulation run spans 500 years, represented as 500 time steps in each series.

Figure 1(a) illustrates a conventional way of storing such data. Time is an additional column in the table. For each simulation run, i.e. for each combination of $diff_H$ and $diff_V$, 500 records represent time steps of the series. The scalar dimensions of the records need to be duplicated for each time step. If, however, we allow some columns to contain curves, we get a different model, shown in Figure 1(b). We have now one record that contains the time series $Tropicaltemperature(t)$ as a curve, as opposed to having 500 records with $Timestep$ and $Tropicaltemperature$ attributes. Duplication of scalar dimensions is not necessary. We have substantially reduced the number of data records and, simultaneously, increased the complexity of the data model. We did not lose any of the data; to the contrary, we have gained additional information. Now values from a single simulation run are grouped into a single record. All curves populating one column in Figure 1(b) constitute a *family of curves*.

Formally speaking, this data can be represented by a data model consisting of m independent variables (the simulation parameters) and n dependent variables (simulation results). The independent variables have scalar values and can be expressed as $x = [x_1, \dots, x_m] \in I$. Here I denotes the set of all possible combinations of values of independent variables, representing all simulation runs. Dependent variables are functions of the independent variables. When a value of a dependent variable k is a curve (data series over t) $f_k(x, t)$, it contains o data elements, one for each value of $t \in t_1, \dots, t_o$. A family of curves is then a set of curves for each possible value of x , $f_k(x_i, t) | \forall x_i \in I$.

If dimensions can be not only scalars, but also curves, then we can improve the analysis significantly. This data model has proven itself in several case studies we have done in different fields, for example traffic surveillance [14], optimization of Diesel fuel injection [14], timing chain drive design [13], and medical data [15].

4. THREE LEVELS OF COMPLEXITY IN INTERACTIVE VISUAL ANALYSIS

Interactive visual analysis is an iterative process. It usually starts with a simple analysis of the original data to gain overview. Then, for a more advanced analysis, the analyst needs to use more complex procedures and can combine findings from earlier stages of the analysis. The need for a more advanced analysis arises as information should be extracted which is more difficult to access. At a certain point it becomes difficult or even impossible to find features of interest by analyzing the original data only using conventional IVA methods. More advanced and complex interaction possibilities are required. Alternatively, or perhaps in addition to that, the data can be enhanced by computing aggregates or first derivatives of time series, for instance. Figure 2 illustrates our view on the three levels of complexity in interactive visual analysis. We exemplify those three levels using the climate simulation data set introduced in Section 3.

In a coordinated multiple views system, the *first level* (smallest circle, left in Figure 2) can be interpreted as simple linking and brushing with one brush. Simple brushing includes selecting a rectangular region in scatter plots, a range of an axis in parallel coordinates or using a line brush [14] in the curve view. The user interactively selects some items in one view, and the selected data subsets are highlighted consistently in all views. Then a different set of items (another *feature of interest*) is brushed and the highlighted patterns in the linked views are studied. The user repeats this process, engaging in an iterative IVA loop. This is sufficient in many cases, e.g. when identifying diffusivity parameters that lead to high temperatures (Figure 3), but we cannot formulate more complex queries. We cannot identify the driest cases, for instance, because we would need to simultaneously brush high temperature and low precipitation.

On the *second level* (larger circle in the middle of Figure 2) more views are used and brushes can be combined with logical operators. This makes answering such questions possible, and it is at large the current state of the art [17]. Several brushes can be defined in the same or in different views. Brushes can be combined using logical operations (AND, OR, NOT). The data selected by the composition of the brushes are highlighted in all views. Brushes can be composed via a *feature definition language* [6], or in an iterative manner [14]. This enables drill down (AND and NOT) or

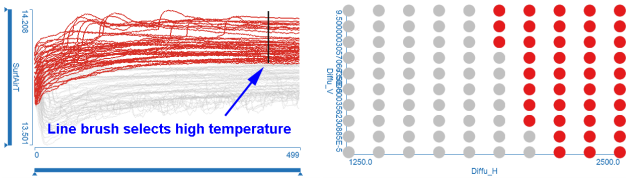


Figure 3: An example of a level one analysis with linking and brushing. High temperatures at a later point in time are selected with a line brush. The corresponding model parameters are highlighted in the scatter plot.

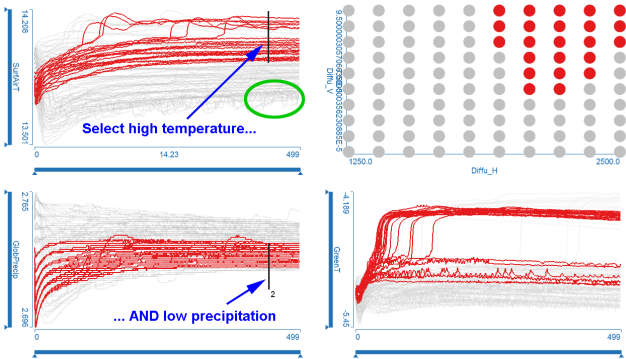


Figure 4: An example of a level two analysis. A composition of two brushes highlights the droughtiest cases. High temperatures (top left) and low precipitation levels (bottom left) at the same point in time are brushed. The scatter plot displays model parameters $diff_H$ and $diff_V$. Greenland temperature is shown in the bottom right. However, level two analysis is not powerful enough to select the jagged lines marked by the green ellipse.

broadening (OR) of the selection. The droughtiest cases can be found easily with composite brushing (Figure 4). This is still not sufficient for more detailed analysis of complex data. There are many more analysis goals that cannot be achieved by the use of those methods, for example, identifying cases which have rising temperature at some point in time, or finding temperature curves of some specific shape.

We can see some jagged curves marked with the green ellipse in Figure 4. There are also some faintly visible jagged curves in the precipitation plot. Is there any correlation between them? Do they belong to the same simulation runs? How do we select them? One possibility is to compute the first derivatives of curves. Extreme (either positive or negative) values of the first derivatives can be brushed to select jagged curves. The procedure is illustrated in Figure 5. It must be mentioned that the first derivative of discretely sampled data is approximated by finite differences, and the data often needs to be smoothed before differentiation to compensate for frequency amplification due to derivation.

We observed that there is a certain duality in complex analysis tasks. One option is to use complex interaction methods and visualizations, and retain data in its original form. Com-

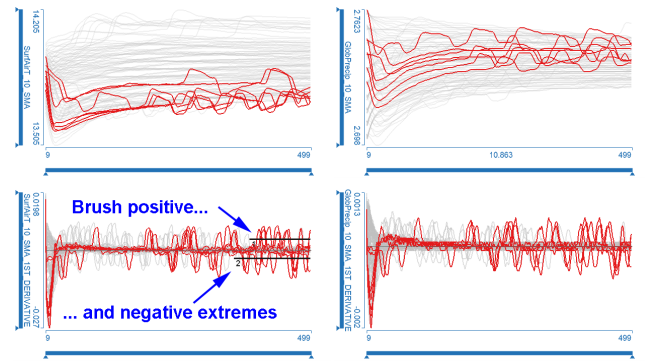


Figure 5: An example of a level three analysis. The plots show temperature and precipitation (top) and their first derivatives (bottom). Two line brushes select extreme temperature derivatives near the end of the simulation run. The jagged temperature curves become visible, including two that were hidden in Figure 4. The highlighted precipitation curves are similarly jagged, indicating correlation.

plex interactions include, for instance, the angular brush in parallel coordinates [9] and the angular line brush in the curve view [16]. Conversely, new data can be synthesized during the analysis. This derived data can often be analyzed with simpler interaction techniques. Both approaches have advantages and disadvantages; none is universally preferable. There is usually a learning curve associated with complex visualization and interaction techniques before analysts can use them effectively [12]. On the other hand, the meaning of derived data is not always intuitive. Depending on the task and the analyst's experience and background one or the other method is preferred. The interactive visual analysis framework needs to provide support for both.

The combination of complex interaction and on-demand data computation constitutes the *third level* of visual analysis (rightmost circle in Figure 2). In order to compute data for the next iteration of the analysis, the analyst should not need to interrupt the analysis session and use some different tool to perform computations. Such detours can significantly hinder the analysis. Quite the contrary, the computation must be tightly integrated into the IVA framework. Therefore, while the data in levels one and two remains static throughout the analysis session, it becomes dynamic in level three. Data changes as a result of the on-demand computations—for each computation, a new dimension (a column in the table in Figure 1(b)) is added to the data.

Note that each of the three circles in Figure 2 encloses the previous ones from the lower level(s), indicating that a seamless switch between circles (levels of analysis) is possible.

5. ANALYSIS OF FAMILIES OF CURVES

In this section we illustrate the above described principles in the context of simulations in automotive industry. Current emission regulations and efficiency requirements for modern car engines lead to very complex designs. Engineers have to deal with many, often contradicting, parameter settings. The use of simulation is unavoidable in modern engine de-

sign. Advances in simulation and computation technology have made multiple simulation runs possible. The idea is to run simulations for the same simulation model with different sets of control parameters. Interactive visual analysis is a perfect method to analyze and explore the resulting data sets. We have analyzed two important systems (Diesel fuel unit injector and variable valve actuation) by means of interactive visual analysis. A complete description of the two case studies is out of scope of this paper. Interested readers can find additional information in the referenced literature [7, 19]. We rather selected four of the most interesting analysis patterns detected. These patterns are applicable to virtually all domains where families of curves are analyzed.

Variable valve actuation (VVA) [19] is an active research field in automotive industry. The operation of four-stroke internal combustion engines consists of intake, compression, power, and exhaust strokes. During the intake stroke, the inlet valve opens and vaporized fuel mixture enters the combustion chamber. During the exhaust stroke, the exhaust valve opens and the exhaust gases leave. Conventional systems use a camshaft, where cams open and close the valves at times dependent on the mechanical construction of the cams. Different operating conditions (e.g. load or engine speed) require different valve opening and closing timings to achieve optimal efficiency and emission. In engines equipped with a VVA mechanism, the valves' opening and closing times, as well as the valve lift can be controlled based on the load and engine speed. We explore the design of a hydraulically supported cam operated VVA system. Such a system has a camshaft with cams, and an additional hydraulic mechanism which opens the valve independently of the cams' position. Approximately 600 simulation runs were computed with different parameters and the ensemble data was analyzed.

Besides valve actuation, the *fuel injection subsystem* [7] is another engine component that has crucial impact on emissions and performances. Currently, the two most important types of injection systems for Diesel engines are the common rail and the unit injector systems. Common rail systems keep fuel pressurized to the injection pressure in a fuel rail which feeds the cylinders. The rail is common to all cylinders. In contrast, unit injector systems have the high pressure fuel pump integrated with the injector. There is one injector/pump unit per cylinder in the cylinder head assembly. We analyzed [16] the Delphi E3 Diesel Electronic Unit Injector (EUI) [7], an advanced unit injector with two independent, fast-response precision actuators that can change the injection pressure level and adjust the fuel delivery timing and duration, allowing a very flexible choice of fuel injection characteristics. 2880 different simulation runs were computed and analyzed in the case study [16].

5.1 Aggregates and Thresholds

Some of the typical goals in the analysis of families of curves can be effectively tackled by computing some aggregate of each curve in the family. The most important aggregates from the IVA perspective are minimum, maximum, arithmetic mean, percentile (which also includes median), and integral. These aggregates are easily computed, yet they convey useful information. Aggregates are scalars that represent an entire curve, significantly reducing data complexity. Thereby, standard visual analysis tools that are avail-

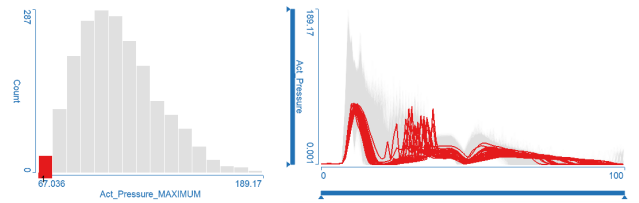


Figure 6: Items with the smallest maximum pressure are brushed in the histogram.

able for scalar data can be used in the exploration of families of curves. This possibly requires less complex displays and simpler interaction, too.

The most often used aggregates in engineering are certainly minimum and maximum. It is interesting that analysts are often interested in curves with the smallest maximum or the largest minimum values. Such constraints describe curves that do not go over or fall below specific *thresholds*. The thresholds may represent desirable cases or outliers that need to be avoided. In Figure 6, the maximum values of the pressure in the valve actuator were computed. The reduced data complexity allows selecting curves with the smallest maximum in the histogram—a view much simpler than the ones normally used for families of curves.

5.2 Exploring Slopes

In many fields, a typical task when analyzing families of curves is finding curves that rise or fall at a certain point in time. This is not easily possible with only level one and two IVA. It would require stopping the analysis, precomputing first derivatives, loading data again and then using another curve view to brush positive or negative first derivatives at the time point. Level three, on demand attribute derivation or advanced interaction, makes this kind of query much easier and faster, without interrupting the IVA process.

During the analysis of the Delphi E3 EUI we want to achieve the high power mode of operation. The injection rate curve has to rise and decrease very steep. In addition, the injection pressure must be as high as possible in order to inject the sufficient amount of fuel.

Before analyzing the control parameters causing the desired slope, we want to make sure there is no second needle opening, an undesirable phenomenon that happens sometimes. The needle is opened once more at the end of the cycle, leading to an unwanted, uncontrolled subsequent injection, resulting in the rapid deterioration of the quality of the combustion. In the data this phenomenon appears as short rising sections in the injected fuel rate curves. There are more than 2800 overlapping curves in the curve view in Figure 7(a), and we cannot see if there are any with rising sections near the end. We can use the first derivative of the injected fuel rate curves to examine such cases. The user simply “orders” the first derivative of the family of curves and a new dimension is created in the data set. A new curve view is opened in the CMV to show the first derivatives (Figure 7(b)). Positive first derivatives are seen at the beginning of the injection cycle—rising curves, as expected. At the end of the injection cycle we can see curves with negative values (falling curves,

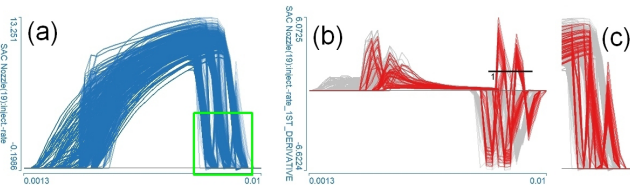


Figure 7: (a) Over 2800 overlapping curves depicting injected fuel rate. There are some curves rising near the end (green rectangle), but they cannot be identified because of occlusion. (b) The user requested the computation of first derivatives. The slopes of interest can be brushed easily in the plot of the first derivatives. (c) A zoomed view of the selected curves with rising parts.

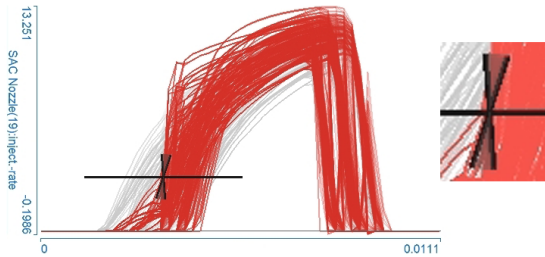


Figure 8: The angular line brush (zoomed on the right) selects curves that intersect it at a given threshold of angles. Steep rising curves are selected.

as expected), and also positive ones. Positive derivatives here are unwanted; those are curves we wanted to identify. We can easily brush them now using a simple line brush.

We can also select curves having a certain slope by enhancing the interaction. The *angular line brush* [16] is proposed as a method for the intuitive brushing of curves based on the angle. As the curve view already has its own type of brush, the line brush, which is proven and well accepted, it seemed as a best solution to improve it to be able to brush curves with a specific slope interval. In parallel coordinates, angular brushing [9] has been proposed as a method to brush lines that have particular slopes. In our case, the user can simply define a range of angles on the line brush. Only curves that cross the line at an angle within that range are selected. Figure 8 illustrates a case from the EUI analysis where steep rising curves are brushed. Although we can allow the angle constraint for arbitrary oriented line brushes, it proved to be most intuitive in combination with either horizontal or vertical ones. For arbitrary oriented line brushes, the user would need to mentally combine the angle constraint with the slope of the line brush in order to brush particular slopes. The angular line brush is used in the original curve view, so it saves valuable screen space. The alternative approach, computing the first derivative and using a simple line brush requires an additional curve view.

Once the first derivative is computed it can be used as input to aggregation. When looking for curves which do not rise or fall significantly, the first derivative can be computed first, and then its minimum and maximum scalar aggregates. The scalar aggregates are depicted in a scatter plot and curves

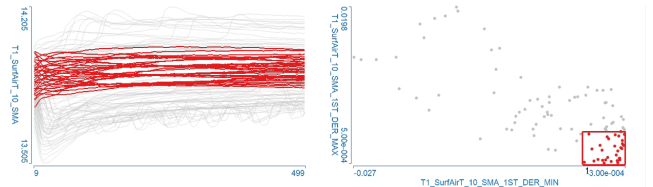


Figure 9: Finding temperature curves that remain “flat”, i.e. do not have steep rising or falling sections. Low maximum and high minimum values of the first derivative are brushed in the scatter plot to select those curves.

having high minimum and low maximum of the first derivative can be brushed now. These are flat curves. Figure 9 illustrates such a case.

Brushing slopes is a premium example of the two identified approaches for improving IVA. We described both approaches, i.e., attribute derivation and using standard interaction methods; versus introducing new interaction methods that work with the original data. They are equally intuitive and both have their advantages, depending on the task.

5.3 Exploring Shapes

In many cases, engineers are looking for curves of certain shapes. Sometimes a combination of several line brushes is sufficient to isolate curves of desired shape [14], but most often a more advanced approach is needed. This problem, too, can be solved by using more advanced interaction, or by computing several specific aggregates and using simple brushes. The well known similarity brush [10, 17] represents the solution using advanced interaction. We propose two ways to perform similarity brushing: the user sketches the shape and then all similar curves are selected; or the user picks one of the curves and all curves similar to that one are selected. Various smoothing methods are available to alleviate the detrimental effect of noise on the performance of the shape recognition algorithm. The tolerance used in curve comparison can also be specified.

An example from the analysis of variable valve actuation simulation is shown in Figure 10. We are interested in a specific shape of the curves: quick rise, a certain span of time while the valve is opened, then quick fall to a given range, and finally smooth closing. In Figure 10(a), the shape is defined using a similarity brush composed of four segments. The similarity thresholds can be specified per segment. This facilitates very precise control over shape. As an alternative, we can compute four scalar aggregates, $r1t$, $r2t$, br , pw , shown in Figure 10(b). The aggregates are visualized in a parallel coordinates view in Figure 10(c) and we can brush desired values for all aggregates. A similar set of curves is selected again (Figure 10(d)). We needed specific aggregates in this case, and an additional view to define the shape via aggregates.

Less complex shapes are often easier brushed by several angular line brushes, or by using line brushes on the first derivative. Using the first derivative for brushing shapes requires an additional view. Curvature, the amount by which a line deviates from being straight, is also proposed as a

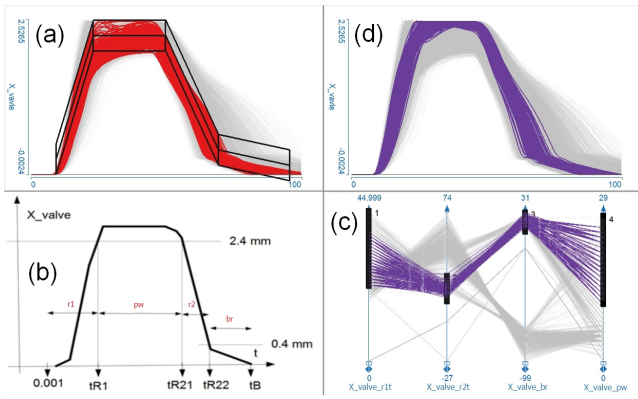


Figure 10: (a) A specific shape of curves is brushed by a similarity brush that consists of four segments. (b) A set of scalar aggregates—timings of characteristic points—that describe shape. (c) Brushing curve shape by selecting ranges of the scalar aggregates in the parallel coordinates. (d) A similar set of curves is highlighted as by the similarity brush.

means of finding curves of certain shapes, although it is useful in only some special cases.

5.4 Cross-Family Correlations

Up to now we have depicted each family of curves in separate views. We have used the CMV system to compare multiple families, and that is not always sufficient. We use an example from the VVA simulation. The hydraulic model of the VVA must be developed based on the energy conservation of the complete system (hydraulic power unit composed of engine and hydraulic pump, the accumulator, and the valve system composed of the valve actuator and the valve itself). The complete hydraulic valve actuator model is simulated. From among the approximately 600 simulation runs, we are looking for the runs where energy consumption is low.

The data contains three families of curves: valve lift, actuator volume and actuator pressure. We can depict them using three curve views. First we select the desired valve lift shape. We can see the corresponding shapes of the actuator pressure and volume curves. The energy consumption depends on both actuator pressure and volume.

We propose the usage of phase diagrams—a plot often used in physics—to visualize two families of curves with a common time variable in a single view. The horizontal and vertical axes of the phase diagram represent the two families of curves. Points are plotted by using the corresponding values of curves in the two families as horizontal and vertical coordinates. A point is plotted for each time step. Successive points are connected by line segments. The result is one line showing values of the two curves. The process is repeated for all other pairs of curves from the two families. This plot often reveals interesting relations between different dimensions of the data set that would be more difficult to discern using only curve views. The phase diagram is also fully interactive. Like the curve view, it supports angular line brush and similarity brush which makes finding and brushing hidden relations simple.

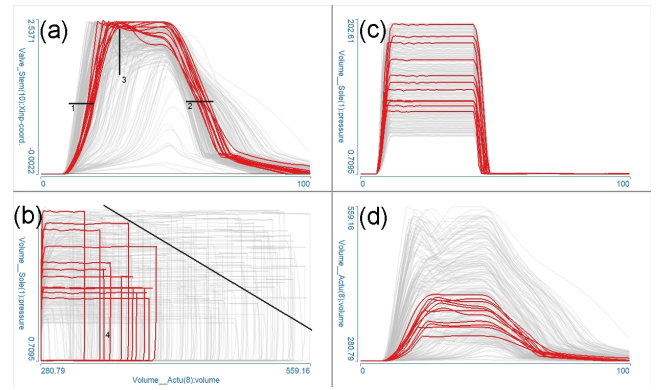


Figure 11: (a) The desired valve lift curves are selected by a combination of three line brushes. (b) The phase diagram simultaneously displays pressure and volume. A simple line brush excludes high energy consumption cases. (c) and (d): Corresponding pressure and volume curves.

In Figure 11, pressure and volume are depicted in the phase diagram. The area outlined by each closed curve in the phase diagram corresponds to the energy used in one simulation run. Once the curves with desired lift shapes are selected, we can refine the selection in the phase diagram by excluding cases with large energy consumption.

6. CONCLUSION

Data sets from many different problem domains contain families of curves, and interactive visual analysis represents a premium methodology for their analysis. We described three levels of complexity in IVA. The first level is represented by linking and brushing with one brush. On the second level, more views are used and brushes can be combined with logical operators. The third level calls for the seamless integration of attribute derivation and advanced interaction.

An advantage of using advanced interaction is that it does not increase the amount of data and the visual complexity, because usually no additional views are necessary. On the other hand, it generates an additional cognitive load, because the user needs to mentally manage the advanced interaction method. New types of analysis tasks may require new, specialized interaction techniques. Designing specialized advanced interaction mechanisms requires a priori knowledge of the expected analysis tasks, which is often not available.

In contrast, attribute derivation increases the amount of data by generating additional synthetic data attributes. The visual complexity is usually increased, because typically new views are necessary to display the derived data. The meaning of the derived attributes may or may not be obvious to the analyst, depending on his or her background. On the other hand, well known, simple mechanisms can be used to interact with the data. Due to the step-by-step approach, with a sufficiently rich set of basis operations, a lot of very different derivations are possible, also ones that were not necessarily anticipated at the time when the IVA system is designed. This opens possibilities to solve unforeseen analysis tasks, too. The series of visualizations generated when

using attribute derivation can be used to discuss findings and communicate the analysis procedure. On the other hand, when advanced interaction patterns are employed, the analysis process is not intuitively captured in the visualizations but it needs to be documented in some other manner.

The advantages and drawbacks of the two essential building blocks, advanced interaction and attribute derivation, are complementary. There is no universally better choice. In open and flexible IVA systems that can be used for a variety of problems, attribute derivation may be preferred. However, in more targeted IVA solutions, where similar problems need to be solved repeatedly, advanced interaction can be more time-efficient for the experienced user.

7. ACKNOWLEDGMENTS

The authors thank AVL-List GmbH for providing the simulation data. Part of this work was done in the scope of the SemSeg project and the K1 program at VRVis. The project SemSeg acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 226042.

8. REFERENCES

- [1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Human Computer Interaction. Springer Verlag, 1st edition, 2011.
- [2] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [3] E. Bauer and A. Ganopolski. Simulation of the cold climate event 8200 years ago by meltwater outburst from Lake Agassiz. *Paleoceanography*, 19(PA3014):1–13, 2004.
- [4] W. Berger, H. Piringer, P. Filzmoser, and M. E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911–920, June 2011.
- [5] E. Bertini and D. Lalanne. Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery*, VAKD '09, pages 12–20, 2009.
- [6] H. Doleisch. Simvis: Interactive visual analysis of large and time-dependent 3D simulation data. In *Proceedings of the 39th conference on Winter simulation*, pages 712–720, 2007.
- [7] G. Greeves, S. Tullis, and B. Barker. Advanced two-actuator EUI and emission reduction for heavy-duty Diesel engines. *SAE Transactions*, 112(3):914–931, 2003.
- [8] H. Hauser. Interactive visual analysis - an opportunity for industrial simulation. In *Proc. of the Conference Simulation and Visualization*, pages 1–6, 2006.
- [9] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proc. of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 127–130, 2002.
- [10] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.
- [11] J. Kehrer, P. Muigg, H. Doleisch, and H. Hauser. Interactive visual analysis of heterogeneous scientific data across an interface. *IEEE Trans. on Visualization and Computer Graphics*, 17(7):934–946, 2011.
- [12] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, editors. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, November 2010.
- [13] Z. Konyha, K. Matković, D. Gračanin, M. Djuras, J. Jurić, and H. Hauser. Interactive visual analysis of a timing chain drive using Segmented Curve View and other coordinated views. In *Proceedings of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization*, July 2007.
- [14] Z. Konyha, K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1373–1385, 2006.
- [15] K. Matković, H. Gan, A. Ammer, D. Bennett, W. Purgathofer, and M. Terblanche. Interactive visual analysis of intensive care unit data: Relationship between serum sodium concentration, its rate of change, and survival outcome. *Proc. of IVAPP*, 2012.
- [16] K. Matkovic, D. Gracanin, M. Jelovic, A. Ammer, A. Lez, and H. Hauser. Interactive visual analysis of multiple simulation runs using the simulation model view: Understanding and tuning of an electronic unit injector. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1449–1457, 2010.
- [17] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A four-level focus+context approach to interactive visual analysis of temporal features in large scientific data. *Computer Graphics Forum*, 27(3):775–782, May 2008.
- [18] R. Peng. A method for visualizing multivariate time series data. *Journal of Statistical Software, Code Snippets*, 25(1):1–17, 3 2008.
- [19] M. Pournazeri, A. Khajepour, and A. Fazeli. An efficient lift control technique in electro-hydraulic camless valvetrain using variable speed hydraulic pump. *SAE Int. J. Engines*, 4(1):1247–1259, 2011.
- [20] J. C. Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In *Proc. of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization*, 2007.
- [21] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the 1996 IEEE Symp. on Visual Languages*, pages 336–343, 1996.
- [22] J. J. Thomas and K. A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. NVAC, 2005.
- [23] L. Tweedie, R. Spence, H. Dawkes, and H. Su. Externalising abstract mathematical models. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 406–412, 1996.
- [24] C. Weaver. Cross-filtered views for multidimensional visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):192–204, Mar. 2010.