Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

VISUAL ANALYTICS FOR SIMULATION ENSEMBLES

Krešimir Matković

VRVis Forschungs-GmbH
Donau-City Straße 11
A-1220 Vienna, AUSTRIA

Denis Gračanin

Department of Computer Science
Virginia Tech
2202 Kraft Drive
Blacksburg, VA 24060, USA

Helwig Hauser

Department of Informatics
University of Bergen
Postboks 7803
5020 Bergen, NORWAY

ABSTRACT

We often simulate multiple variations of the same model – a simulation ensemble – to better understand intricate physical phenomena. The analysis of complex simulation ensembles represents a grand challenge which is approached by both computational and interactive, visual methods. We describe how modern visual analytics helps to analyze simulation ensemble data. A clever combination of computational and interactive methods supports the simulation expert to gain deeper insight into the data and into the physical phenomenon that is represented by the ensemble. An analysis environment that combines interactive visualization and computational analysis provides unique advantages for the exploration and analysis of complex ensemble data. It helps the domain expert to efficiently cope with analysis tasks, in particular when they are only partially defined. In this work, we describe the basics of interactive visual analysis, several approaches to interactive ensemble steering, and means for results quantification and analysis reproducibility.

1 INTRODUCTION

Modern science and engineering have become unimaginable without simulation. Simulation has established itself as a premium mean and as an unavoidable methodology for the study of challenging problems. Generally, two questions are of great interest when working with numerical simulation: (1) What is the simulation output for a certain model parametrization, and (2) which parametrization leads to a desired simulation result. Answering the first question is relatively straight-forward by running the simulation. Answering the second question – inverting the simulation model – is far from trivial, except for very simple models. Realistic models can not be inverted analytically, and alternative approaches to solving this problem are needed.

Simulation ensembles – multiple simulation runs, based on a set of differently parametrized simulation models – represent a possible solution. In addition to helping with finding a desired parametrization, they also support getting a deeper insight in the functioning of the model, and, consequently, in the studied phenomenon. In order to exploit a large amount of usually complex data, which results from an ensemble simulation, a new analysis methodology is necessary. When analyzing ensemble data, we can rely on computational data analysis methods, we can use interactive visualization, or we can combine

both approaches, leading to a visual analytics solution. Computational methods are very powerful and a lot of different solutions have been researched. To deploy them, it is usually necessary to exactly know what to look for. Interactive, visual methods, on the other hand, can be useful for fast and flexible data exploration, potentially revealing insight that is unexpected. They exploit the strengths of human analysts, i.e., their perception, cognition, knowledge, and imagination. Using visualization alone, however, limits the complexity of analysis tasks. A clever combination of interactive and computational methods can be the key to a successful analysis of challenging simulation ensemble data. Visual analytics offers such a combination.

Ensembles occur in various forms, e.g., they can consist of scalar or vector fields (Ferstl et al. 2016), weather forecasts (Sanyal et al. 2010), contours (Whitaker et al. 2013), 3D isosurfaces (Demir et al. 2016), image segmentations (Fröhler et al. 2016), or even volume data (Demir et al. 2014). For example, ensembles are often used for meteorology data, where repeated simulations characterize the uncertainty of weather predictions. Wang et al. (2017) proposed nested parallel coordinates to analyze relations between the high-dimensional parameter space and resulting climate ensembles.

Wilson and Potter (2009) provide an overview of ensemble data characteristics and consequences for visual analysis. They also identify general leading questions, which apply across different application domains. Common approaches dealing with such data are based on complex ensemble members, but they mainly address smaller ensembles (around 50 members) (Demir et al. 2014; Demir et al. 2016; Ferstl et al. 2016).

An intuitive approach to visual exploration of ensemble data is to reduce the complexity by summarizing characteristics via statistical measures (Kao et al. 2001). Potter et al. (2009) presented Ensemble-Vis, which provides statistical aggregation of weather ensembles. They combine different visual representations, e.g., via an overlay, to display multiple measures.

The exploration of simulation input and corresponding output also relates to the field of parameter studies. A recent survey on visual parameter space exploration is provided by Sedlmair et al. (2014). Bergner et al. (2013) presented ParaGlide, a visualization system that allows for exploration and partitioning of parameter spaces for simulation data.

Visual analytics is defined as "the science of analytical reasoning facilitated by interactive visual interfaces" (Thomas and Cook 2005, p. 4). It is a combination of interactive exploration by means of visualization (often using coordinated multiple views) and computational analysis methods. The visual part is essential to integrate the expert in the loop. Coordinated multiple views are a widely adopted visualization method, where we visualize different aspects of the same data using at least two views (Gresh et al. 2000; North and Shneiderman 2000; Roberts 2007). Each data item is depicted in both (all) views. An interactive selection of a subset of data points in one view (called "brushing") highlights the same subset of data points in the other view(s). This consistent emphasis of the same data points across all views is called "linking" and emphasizing certain data points in a visualization, for example by a consistent coloring scheme, is called "focus+context" visualization (Hauser 2006).

Figure 1 illustrates *linking and brushing* using two views. The view on the left shows two control parameters of an ensemble. The parameters are varied by means of a Sobol sequence (Sobol 1976). The scatter plot on the right shows two simulated attributes from a variable valve actuation system of a car engine. The user has interactively selected a subset of the parameter space (on the left) – the corresponding runs are highlighted, accordingly. In this example, only two views are used; usually, a more advanced analysis will depend on several (linked) views.

Already a simple solution (like two views with linking and brushing) represents a significant improvement over static plots or the manual inspection of single runs. More views and advanced interaction solutions, including multiple brushes, make the exploration even more powerful. As the number of simulation model parameters rises, the question emerges of how to vary them, and how to create an ensemble in a reasonable time. Ensemble steering makes it possible to initiate new simulation runs during the exploration. The ensemble grows as we explore it. Adding computational methods to the workflow further increases the

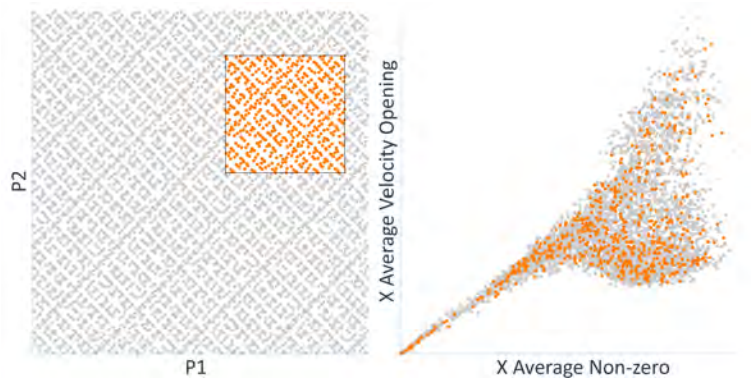

Figure 1: *Linking and brushing*. Two views show the same data. The interactive selection of a data subset in one view (called *brushing*) leads to a consistent highlighting of this subset in all views (called *linking*). Changing the selection interactively leads to an immediate update of this highlighting in all views.

efficiency of the analysis. It also helps with the reproducibility of results and their comparison, as well as with the evaluation of findings. Usually, neither approach alone is sufficient and only a clever combination of interactive visualization and computational analysis can solve the most-challenging problems.

Throughout this paper, we use ComVis (Matković et al. 2008a), a coordinated multiple views (CMV) tool developed at the VRVis Research Center, Vienna, Austria (http://www.vrvis.at). We use two data sets to illustrate how modern visual analytics can help to analyze simulation ensemble data: a common rail Diesel injection system example (Matković et al. 2015), and a variable valve actuation system (Matković et al. 2017). The common rail injection system is the standard injection system for Diesel car engines (Boecking et al. 2005; Boehner and Hummel 1997). It uses an electronic control unit to control the fuel delivery, injection timing, injection pressure, and rate of injection for multiple injection strategies. The goal is to have the level of performance and driving comfort similar to those of gasoline-powered models while reducing fuel consumption and lowering exhaust emissions.

The common rail system uses a high-pressure rail that is common to all cylinders. The high pressure allows for the precise injection of the fuel into the cylinders using electronically controlled actuators that open and close the injectors at least several hundred times per second. There are five control parameters in our example: dT_v (time interval of the injector valve opening and closing), dT_p (time interval of modulated pressure increase on the injectors inlet), P_{low} (low pressure on the injector inlet), P_{high} (high pressure on the injector inlet), and T_{vl} (injector valve opening time). We use five values for the parameters dT_v, dT_P, P_{low}, and P_{high} and seven for T_{vl} resulting in 4375 simulation runs, i.e., ensemble members.

Variable valve actuation (VVA) is an active research field in the development of four-stroke engines. A precise control of the opening and the closing of the intake and the exhaust valves of an engine cylinder is essential for an optimal engine operation. In contrast to the traditional, cam operated valve systems, the VVA makes it possible to change the shape and timing of the opening and closing of the valves. We deal with a hydraulically-supported, directly operated system that has no cam at all. Such a flexible system can ensure the variable feeding and dissipation of the gases involved in the combustion process. We vary nine different parameters in this example.

2 INTERACTIVE VISUAL ANALYSIS OF ENSEMBLE DATA

In general, there are many different types of simulation solvers. Depending on the studied phenomenon, the inputs to the solver can have different types, and the solver outputs are also of different types. In this work, we focus on simulation runs, where a single run expects a set of scalar values as input or control parameters, and for each specific set of control parameters simulation results are computed. The results can be scalar values or of more complex data type such as time series, for example. If there are time-dependent results,

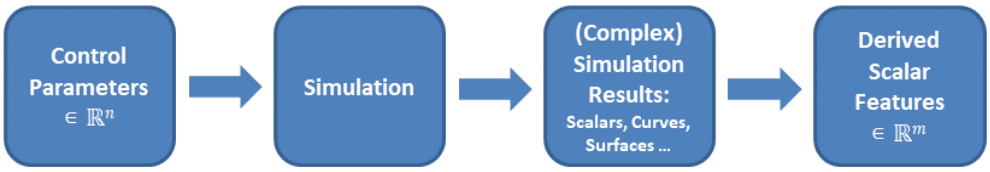

Figure 2: Basic simulation model. For a set of scalar control parameters the simulation solver computes a variety of output values (scalars or more complex outputs, such as time series). Furthermore, scalar values are often derived from complex simulation results to support the analysis.

we can compute various scalar aggregates or other derived values, such as the minimum or maximum of the time series, to support the analysis. Figure 2 illustrates this model for a single simulation run.

The simulation can be seen as a function S that maps the control parameters $\mathbf{x} = (x_1, \ldots, x_n)$, i.e., a control data point in $\mathbb{R}^n$, to the output values $\mathbf{y} = (y_1, \ldots, y_m)$, i.e., an output data point in $\mathbb{R}^m$, where n is the number of control parameters and m is the number of outputs:

$$\mathbf{y} = S(\mathbf{x}) \tag{1}$$

A simulation ensemble E is a set of pairs of such data points $(\mathbf{x}^i, \mathbf{y}^i)$, i.e., $E = \{(\mathbf{x}^i, \mathbf{y}^i) : i = 1, 2, \ldots, r\}$, where r is the number of control parameter vectors used as input to the simulation S. While we are focusing on deterministic simulations, where for a given $\mathbf{x}$ a simulation provides a single, unique value $\mathbf{y}$, there is a significant body of work on using visual analytics and stochastic simulations (Luboschik et al. 2014; Schulz et al. 2011) where multiple runs (replications) for the same $\mathbf{x}$ may produce different values of $\mathbf{y}$.

Since we also consider cases where the simulation results in output values that are not only scalars, but also of more complex data type, we need to refine the above model, accordingly. One solution is to replace non-scalar output values with several derived scalar features. If, for example, the simulation computes a force on a crankshaft as a function of time, the minimum and maximum forces might be sufficient for certain analysis tasks. If we strive for a deeper insight, scalar features might not be sufficient – we may wish to study the non-scalar data directly in the analysis. Then, for each output data point $\mathbf{y}^i$, there are some dimensions y^i_k that are data series and we have a separate set of "sub-points" with own length and number of dimensions. Although such a data model can be arbitrarily extended – to surfaces, for example (Matković et al. 2009; Piringer et al. 2012; Cibulski et al. 2017) – we focus on 1D data types (time series, "curves") in this work, only.

In order to simulate an ensemble, we have to decide which points $\mathbf{x}$ from the parameter space will be used. Domain knowledge plays an essential role here. Usually, the domain experts know about plausible ranges of input parameters. Once the ranges are set, the number of points has to be determined. Methods for the design of experiments (Montgomery 2001) deal with the appropriate sampling of the parameter space. We can choose a full factorial approach, where all possible combinations of the chosen input parameters will be computed, or a more advanced sampling method such as a Sobol sequence (Sobol 1976), i.e., a quasi-random low-discrepancy sampling of the parameter space. The left scatter plot in Figure 1 shows two input parameters sampled by a Sobol sequence. A scatter plot in the case of a full factorial approach looks much sparser (Figure 3). Even if we decide for ten variations of values for two parameters (which might be prohibitive in the case of ten parameters, for example), there will be only 100 points in the scatter plot. However, in this case each point in the scatter plot represents many simulation runs, varied according to other parameters, not shown in this scatter plot. Figure 3 shows such a case. The ensemble is created by varying the values for five parameters, four of them with five values and one with seven, resulting in $5 \times 5 \times 5 \times 5 \times 7 = 4375$ simulation runs. There are 25 (5×5) combinations of values for the two parameters shown in the scatter plot. The values for the other three parameters, not shown in the scatter plot, have 175 ($5 \times 5 \times 7$) combinations. Therefore, each point in the scatter plot represents 175 simulation runs (for different combinations of values for the three parameters not shown in the scatter plot).

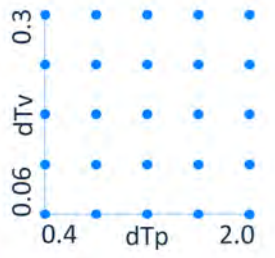

Figure 3: A scatter plot showing $5 \times 5 = 25$ pairs of values for two out of five parameters, dT_v and dT_P. Each of the 25 points in the scatter plot indicates $5 \times 5 \times 7 = 175$ combinations of values for the other three parameters. The ensemble has $5 \times 5 \times 5 \times 5 \times 7 = 4375$ simulation runs, a full factorial sampling of the values for five parameters.

Once the ensemble is computed, the interactive visual exploration and analysis can start. Figure 4 shows all necessary components for a system that enables such a study. The *Design of Experiment* component is responsible for computing parameter variations. The *Simulation* component computes the results per set of parameters. The results are aggregated, if necessary, and the *Interactive Visualization* component shows control parameters, simulation results (scalars and curves), as well as the aggregated values. In case of complex results (curves), it is not always possible to estimate all necessary data derivations prior to the analysis. Basic operations on curves, such as the computation of the minimum or maximum, will usually be required, but some more-advanced data derivations may become necessary during the analysis. Therefore, it is essential to also allow for on-the-fly data aggregation and derivation. The analyst needs a possibility to initiate data aggregation during the analysis. Without such a possibility, the analysis session would be stopped and restarted after data computation, hindering the analysis significantly.

For the interactive visual exploration and analysis, a solution with coordinated, multiple views is usually used. When designing such a system, special care is needed to the trade-off between flexibility and usability. It is certainly good to provide possibilities for configuring many details, but without useful default settings such a system will be complicated to use. In our experience, domain experts from various domains – we have collaborated with engineers, geologists, medical experts, and traffic experts, for example – prefer to

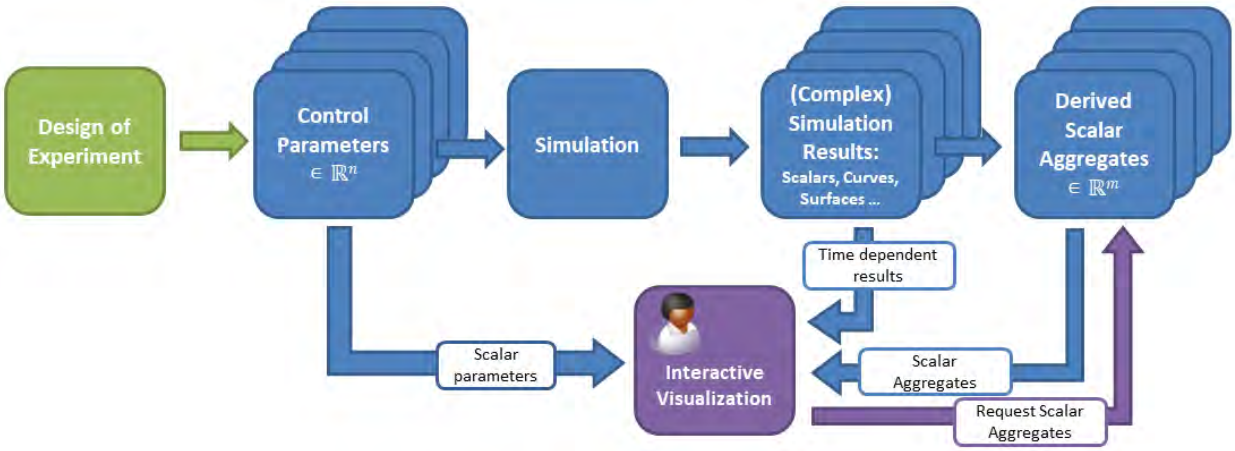

Figure 4: Interactive visual analysis workflow for ensemble simulation data. The *Interactive Visualization* component has a central role for showing, exploring, and analyzing control parameters, simulation results, and aggregations. It is essential to also support on-the-fly data derivation and aggregation.

have control parameters on the left, and outputs on the right (probably related to the left-to-right reading order in the west). Keeping a consistent view layout helps to build a mental model of the analysis setup.

Figure 5 shows a snapshot from an analysis session. Here, the ensemble describes a common rail Diesel injector (Konyha et al. 2006). The engineer configured the views to have control parameters on the left, the needle lift curves (the main point of interest) in the middle, and some other output values on the right. The analysis starts with the user brushing a subset of runs in a view. In this case, he brushed a desired range of two scalar outputs using a rectangular brush in a scatter plot on the right. Then, the user drilled down further by brushing desired curves in the middle using the line brush (Konyha et al. 2006), selecting all curves that cross the brush line. Making such a selection using an SQL command, or inspecting the runs one by one would require significantly more time. The parallel coordinates view on the lower right (Inselberg and Dimsdale 1987) shows curve aggregates that were computed on-the-fly. Below, the table shows details for the selected runs. The analyst can inspect all data available for the brushed subset. The subset can also be exported.

3 INTERACTIVE ENSEMBLE STEERING

The initially computed ensemble is not always sufficient for a successful analysis. There are several reasons for refining an ensemble. It is not always possible, for example, to properly set all parameter ranges in advance. Sometimes, significant portions of the sampled parameter space result in undesired output and should be excluded. Further, if there are, for example, eleven parameters to vary and we would like to have ten variations of each of them, we would need 10^{11} simulation runs. Running so many simulation runs is usually not possible due to time constraints, even when using supercomputing.

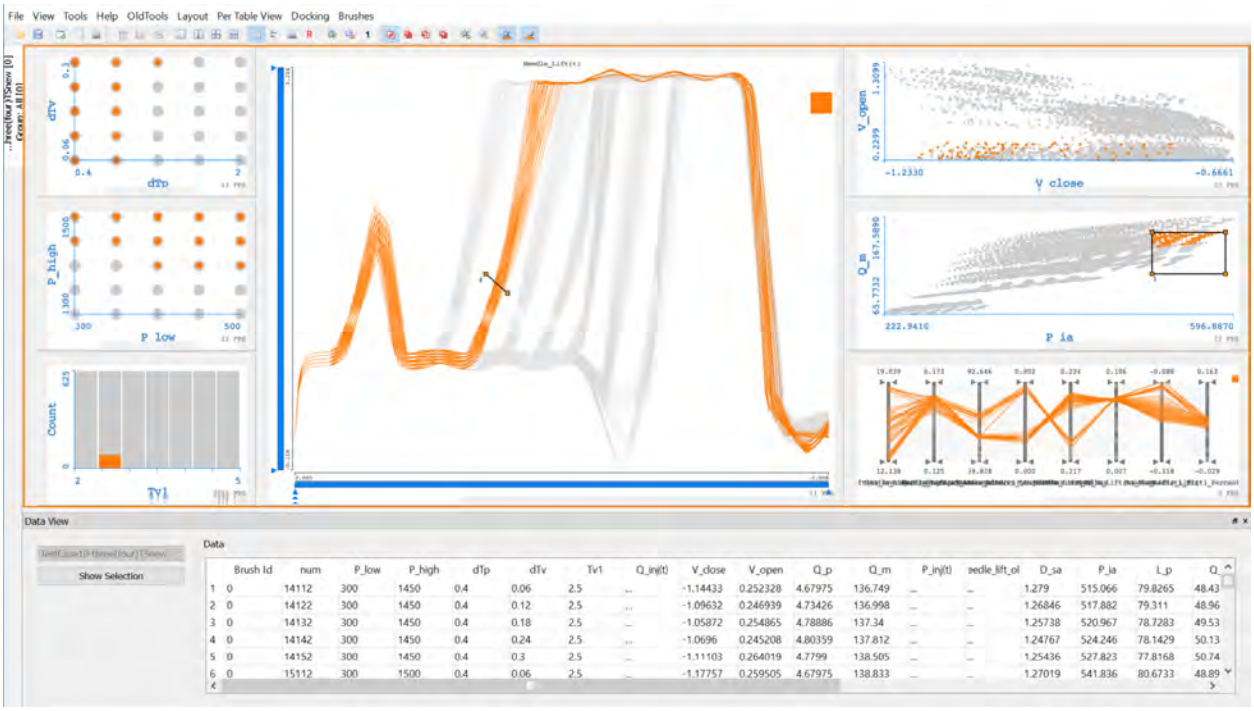

Figure 5: A snapshot from an analysis session. The views on the left show five control parameters (full factorial sampling). The curve view in the middle shows the needle lift of a Diesel engine injector. The views on the right show scalar outputs in the scatter plots and scalar features of the curves that were computed on-the-fly. The selected (brushed) data are shown in orange, while the context (rest of the data) is shown in gray. The table shows details for the selected runs.

In order to cope with the delicate challenge of choosing a proper set of simulation parameters, we exploit the option to initiate new runs from the analysis when necessary. The idea is to first sample the parameter space coarsely, and then, when the user identifies an area of interest, to initiate additional runs only where it is needed. This approach is called interactive ensemble steering (Matković et al. 2008b).

Figure 6 shows the necessary connection between the interactive visualization component and the design of experiments component. Figure 7 shows a scatter plot of two input parameters before and after such an interactive refinement. Here, the analyst has realized, after a first visual analysis, that the initial range for the $I2$ parameter was too small and that she needed more runs for specific ranges of the parameters.

In the case of very complex models we usually start with a simplified model. On top of the model parameters, the model is refined during the analysis. This means that there will be new parameters to consider. In standard steering, "only" new rows are added to the table – we parametrize an existing model with new values.

If we change the model itself, new columns are added to the data, representing new control parameters and new output values. Such a change is computationally more demanding than adding new rows. In the case of any extension of the data, the interactivity clearly depends on the complexity of the simulation and interactive ensemble steering is only feasible if a single simulation run can be computed relatively quickly. We usually initiate tens or hundreds of new runs in one step – if these cannot be computed quickly, the

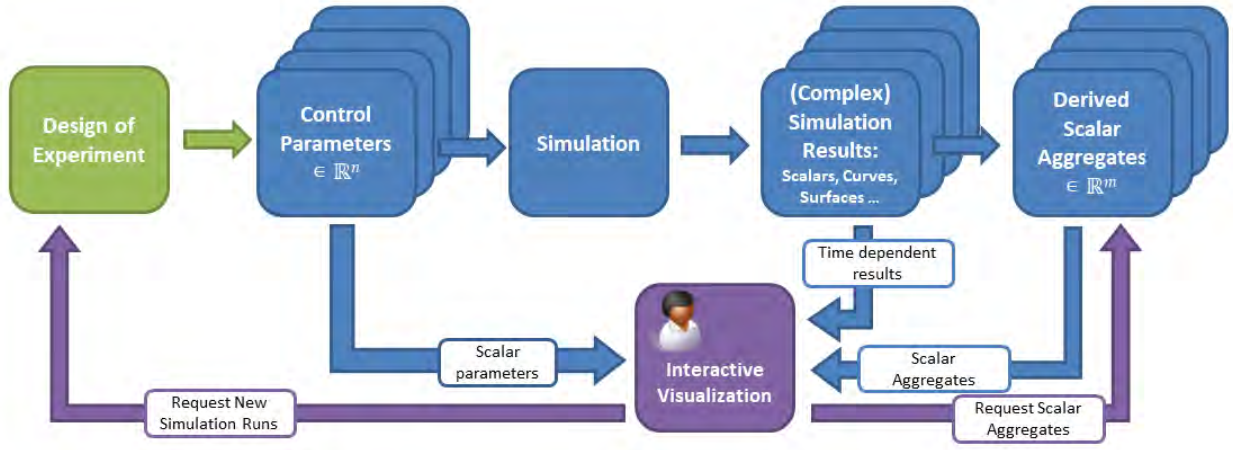

Figure 6: Interactive visual steering is possible due to the connection between the interactive visualization and the design of experiment. The user can request additional runs during the analysis.

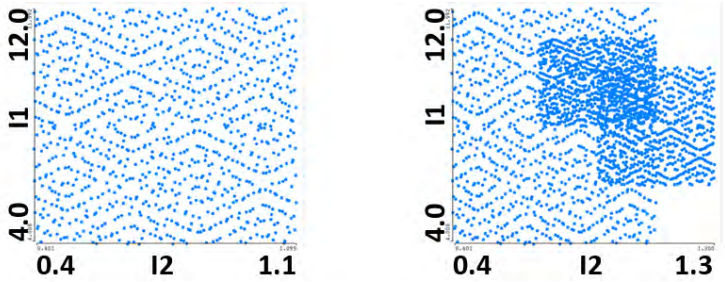

Figure 7: Refining an ensemble by interactive visual steering. After the first visual analysis of the initial ensemble according to the parameters, as shown in the scatter plot on the left, it became clear that additional simulation runs were needed for two particular regions of the parameter space. The scatterplot on the right shows the updated parameter space of the ensemble after the refinement.

potential value of performing such a refinement drops, of course. The new runs are then added to the system as soon as they are computed (there is no need to wait for all runs to finish).

There are three loops in the interactive ensemble steering workflow:

- The *inner analysis loop* represents the iterative process of interactive visual exploration and analysis. The expert uses linking and brushing in coordinated, multiple views to study a particular ensemble.
- The *ensemble refinement loop* is used to initiate new parameterizations of the current simulation model. This leads to additional samples from the parameter space and new rows in the data table.
- The *model refinement loop* is used to refine the simulation model itself. This is the most demanding operation as the data structure is changing. New columns are added to the data table.

When dealing with a complex model with many parameters, it is not easy to identify the areas in the parameter space that have to be refined. Hybrid visual steering (Matković et al. 2014) combines automatic optimization with interactive visual steering. The main idea is to use an automatic optimization based on a comparably simple regression model of the ensemble in order to approximate the desired operation point. If we are looking for the lowest consumption of an engine, we can compute a regression model of the ensemble and use it to find a point close to the optimum automatically. This identifies a new target area to explore in more detail.

Since the regression model is built from scalar control parameters, scalar outputs, and scalar aggregates of complex outputs (they are not taken into account directly), the computed point, approximating the actual optimum, should be used as a guidance for further exploration. We usually start with a new run in the computed point to see what the actual simulation model computes there.

Additionally, it is useful to compute simulation runs for additional points in the neighborhood of the point. The neighboring points are sampled from the n-dimensional control parameter space around the computed optimum approximation. All these actions are initiated from the interactive visualization during the analysis process. In order to do so, the regression model building and the automatic optimization components have to be added to the system. The regression model building is requested from the visualization and the regression model can be built for all data points or for a subset only. Target values are also specified from the visualization. Figure 8 illustrates the hybrid steering workflow and its components.

This rather complex interactive process required some customized views. We propose the optimization constraints view, which is used to set up optimization constraints and to depict different optimum values computed during the analysis session (Figure 9a).

We also propose a way to depict the precision of the regression model. For each point in the ensemble we compute output values using the regression model. We show two values per point and connect them with a line (Figure 9b). Thirdly, in a deviation plot we depict all regression points in the origin. The deviation amount and direction for each point is visible now, and points of a certain deviation amount and direction can be brushed (Figure 9c).

Based on these exploration and analysis principles, we support a rich set of abstract tasks that are characteristic for the interactive visual study of ensemble data and hybrid steering.

Table 1 describes the tasks. There are three groups of tasks, *Explore and Analyze an Ensemble*, *Ensemble Setup and Refinement*, and *Approximating an Optimum with a Regression Model*. Each group contains three tasks.

4 ANALYSIS REPRODUCIBILITY AND RESULTS EXTERNALIZATION

The interactive analysis process described so far is qualitative in its nature. The user sees different patterns and gets an insight into the simulated phenomenon. Communicating and reproducing such qualitative results is not easy and hinders a wider acceptance of the analysis. VisTrails (Silva et al. 2007) is an example of an open source provenance-management system. The provided infrastructure enables capturing information about how workflows evolved over time.

The analysts must have means of communicating their results, and the results have to be reproducible. It is not enough to say, for example, "I brushed some values in the lower right part of the scatterplot".

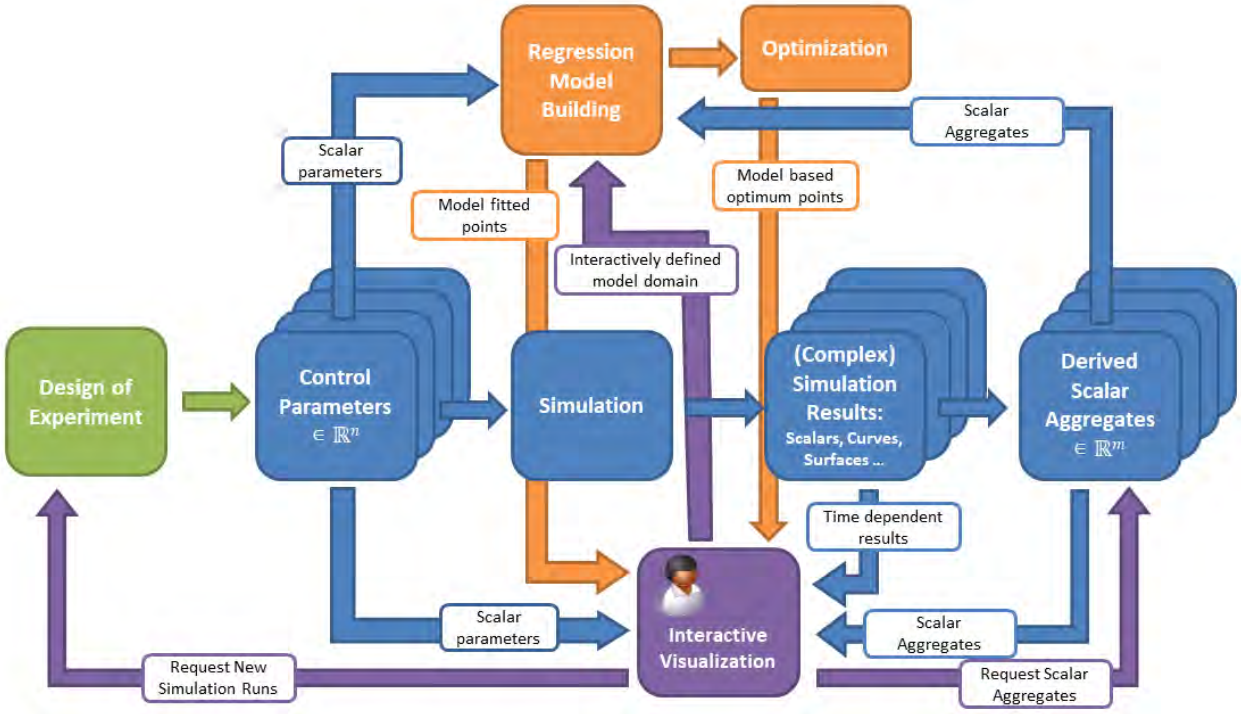

Figure 8: The hybrid steering integrates automatic optimization based on regression models with interactive exploration. All computational actions are initiated and configured from the interactive visualization. Such a system enables a tight interplay between computational and interactive analysis. The computational results are used as guidance for the interactive exploration.

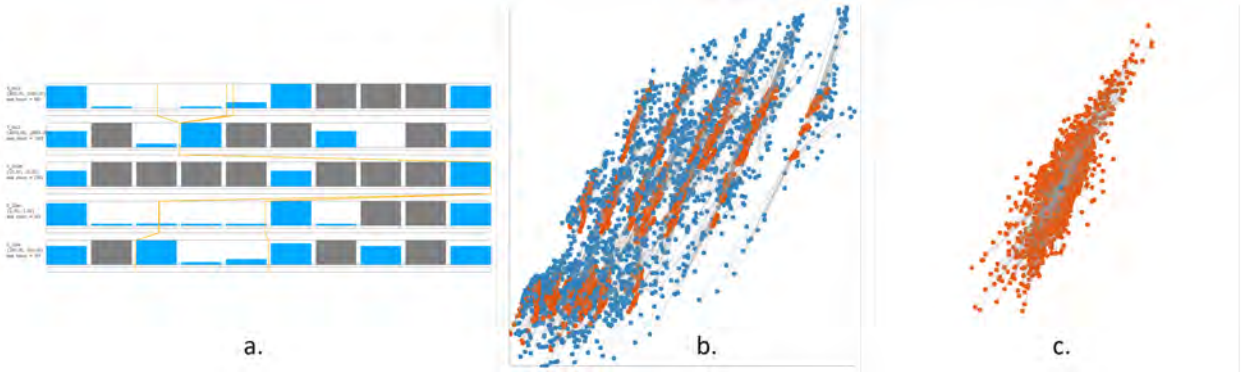

Figure 9: The hybrid steering approach requires new views. (a.) The optimization constraint view is used to set up optimization parameters and to depict computed optimum values. (b.) Original points in orange and regression-model based points in blue are shown for each stimulation run in this 2D projection of the parameter space. Connecting lines help to identify simulated and regression-based pairs. (c.) The deviation plot shows how each run differs from the regression-based point. All regression-based points are drawn in the origin. Desired deviations are easy to brush.

Table 1: Three groups of tasks for the interactive visual study of ensemble data and hybrid steering.

Group I: Explore and Analyze an Ensemble	
Parameters' Sensitivity	Relate outputs to control parameters and explore their sensitivity
Model Reconstruction	Identify control parameters for a desired simulation output
Comparison	Compare simulation outputs related to different areas of the parameter space
Group II: Ensemble Setup and Refinement	
Initial Parameter Space Sampling	Select regions in the parameter space for sampling
Interactive Refinement	Select regions in the parameter space for resampling (or new sampling)
Initiating Optimization	Choose refinement regions for automatic optimization
Group III: Approximating an Optimum with a Regression Model	
Model Validation	Show the model accuracy across the parameter space
Model Definition	Partition the parameter space and define parts for computing the model
Automatic Optimization	Automatic optimization using the regression model

How can another person brush the same values? How can even the same person brush the same values again in another analysis session? The answer is "Not easily". However, there are several ways to improve reproducibility and to make the interactive, qualitative process more quantitative.

It is possible to structure the brushing space (in order to make the interaction easier to reproduce) and to augment linked views with the quantitative information (Radoš et al. 2016). We can also augment the brushing using statistic measures (Kehrer et al. 2010; Haslett et al. 1991). Further, we can also use well-known methods from machine learning to quantify findings and to facilitate communication of the findings (Matković et al. 2017). Instead of saying, for example, "The curves rise steep", we can express the rise angle exactly, and omit any misunderstanding in interpreting the adjective "steep".

Table 2 shows some of the means for structuring the brushing space and augmenting the linked views. Figure 10 shows some of the described features. In order to facilitate brushing reproducibility, a simple snap-to-grid approach can be used. For example, if we know that we have a ten by ten grid, we know exactly what does it mean to brush the lower right cell.

As it is often desired to select rank-based and not value-based parts of the data, we introduce the percentile grid. The percentile grid (Figure 10, top left scatterplot) draws the lines so that each axis segment contains the same number of items. In the example shown in Figure 10, the vertical axis is divided into five segments, each containing 20% of the data points, and the horizontal axis is divided into ten segments, each containing 10% of the data points. Depending on the points' distribution, some of the grid elements will be wider and some narrower. It is now easy to brush, e.g., the lowest 10% of points on the x-axis.

In addition to the snap-to-grid functionality we propose to automate the brush movement operation. The rationale behind the movement automation is the very basic idea of the brushing. When we move the brush, we observe what is happening in the linked views. Since we know how we move the brush (from

Table 2: A set of possible measures for increasing reproducibility of the analysis process. The measures help to structure interactive brushing and to augment linked views.

Brushing	Linked Views
Structure space (grid lines)	Overlay statistics
Constrain interaction (snap to grid)	Show history
Automate interaction	Show relative values
Change interaction domain (ranked based vs. value based)	

high opening speed towards low opening speed, for example), we can focus on the linked views. The changes there have to be observed in the context of the moving brush.

It is difficult to reproduce exactly the brush movements over and over again. If we define a brush movement and let the system move the brush (in a loop), we can focus on the linked views, while keeping in mind the moving brush.

The linked views can be augmented with descriptive statistics. The parallel coordinates plot shown in Figure 10 shows the mean, median, and middle value for brushed data for each axis. The exact values make it easier to communicate results. The scatterplot on top right in the same figure shows statistics for the linked data. In addition, three curves on the right show how main descriptors change as the brush moves. In this way we can externalize some information and reduce the analyst's cognitive load. The analyst sees how the values change and, by knowing the cause (brush movement), understands the data much better.

The question of how to effectively and efficiently externalize valuable findings so that the subsequent analysis steps can build on them remains a difficult challenge. There is a limited body of work (Yang et al. 2007; Shrinivasan and van Wijk 2009; Lampe and Hauser 2011) that focuses on this challenge.

The quantitative externalization of findings from the qualitative interactive analysis process is genuinely difficult, while many workflows clearly would benefit from solutions that provide results in a quantitative form. One solution is to locally model selected data relations of interest with a linear data model and then externalize the model parameters from this process (Matković et al. 2017).

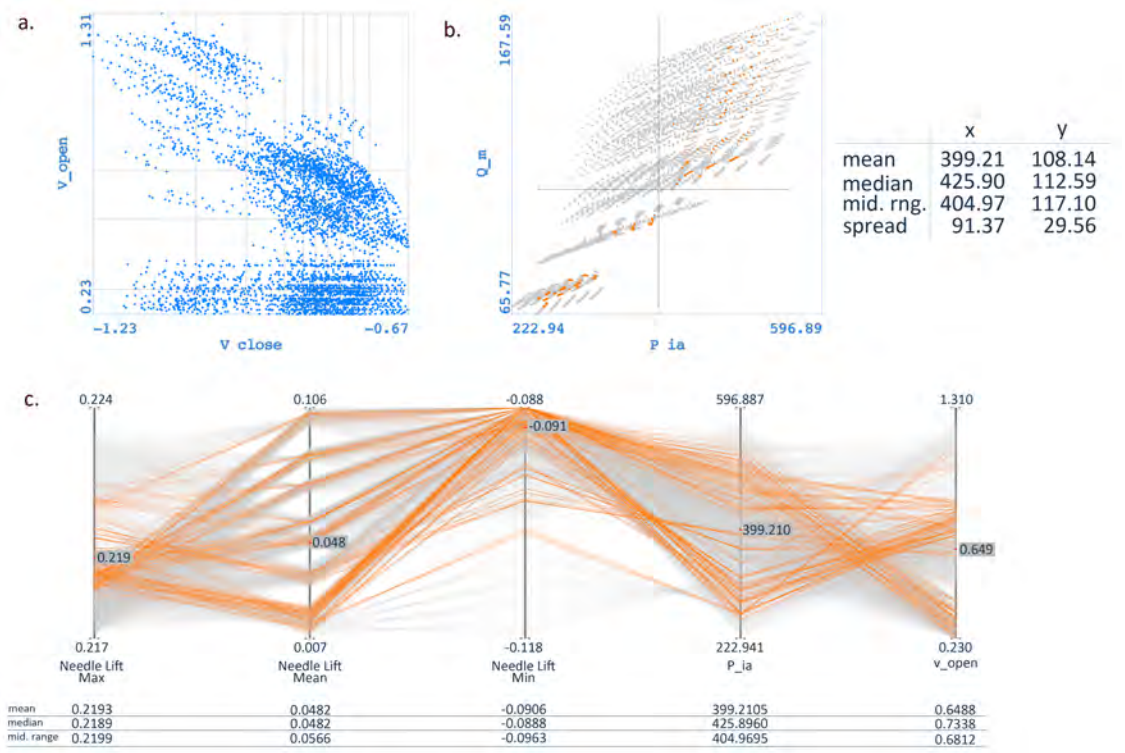

Figure 10: A subset of techniques which improve analysis reproducibility. (a.) Percentile grid divides a scatterplot so that an equal number of points is in each vertical and horizontal strip of the grid. (b.) A linked view augmented with descriptive statistics. The cross hair in the scatterplot shows the span of brushed data. (c.) The parallel coordinates view shows descriptive statistics for the brushed data for each of the axis.

For several reasons, mostly due to stability and simplicity, we focus on the linear local models. While linear models can be too simple for global data approximations, they often provide good local results. Therefore, it is important to enable local model building as well as some kind of model evaluation.

Finally, keeping tracks of models and of analysis results is essential for reproducibility. Figure 11 shows a screen shot of such an analysis. The dialog in the right frame is used to define linear models of an ensemble simulation. The user chooses independent and dependent variables and models to build. The user also specifies the model name. If brush data are present, only brushed data can be selected and used for model building. The tables in the left frame show parameters of previously computed models. The coefficients can be compared to evaluate the models.

Finally, it is nice to support users during the iterative and interactive exploration and analysis process. For this purpose, there has to be a way to store findings, to easily recall the findings, and to explore different possibilities of how to proceed with the analysis from a certain point.

5 CONCLUSION

The tutorial represents the first step only and does not cover all aspects of ensemble visualization and visual analysis. Interested readers are encouraged to explore the topic further. The methods presented here help the analyst to better understand the ensemble data. When dealing with complex ensemble data, the analysts are confronted with many challenges. Interactive visual analysis represents a perfect addition to the well-known automatic methods in data analysis. Adding a human in the loop exploits humans' advantages, and combining them with computational power represents a winning combination.

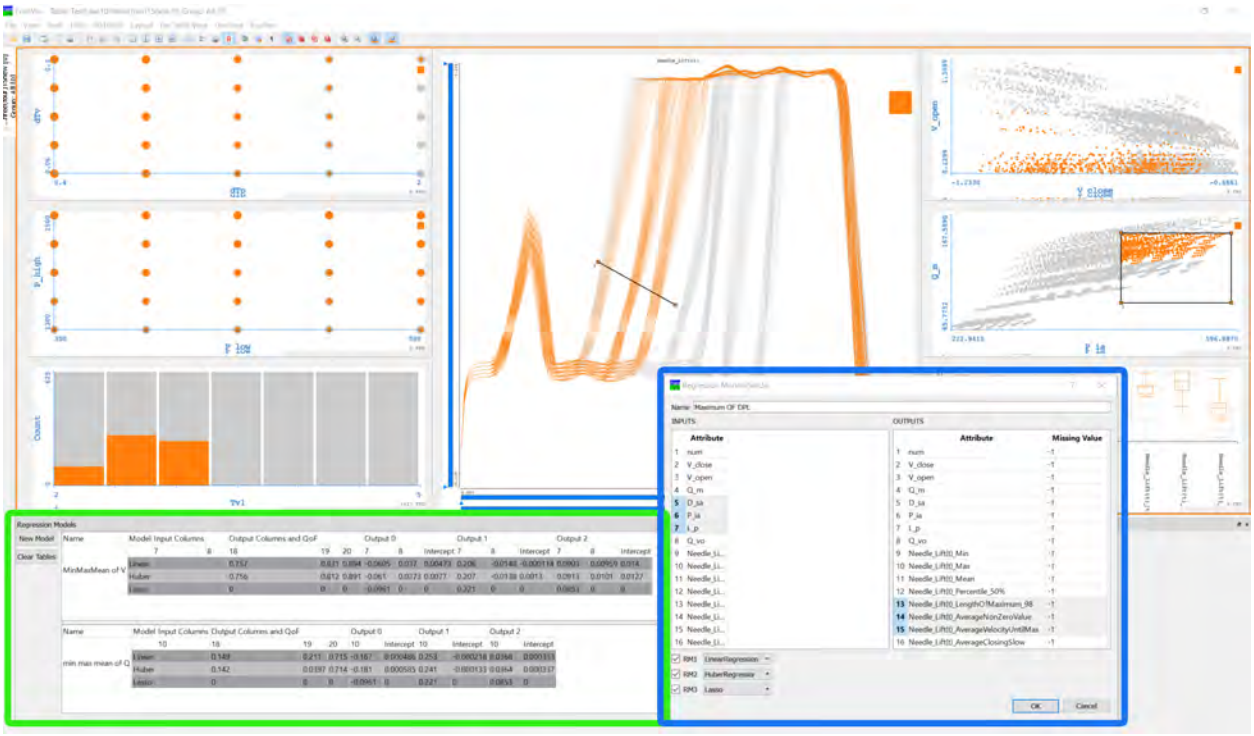

Figure 11: A screenshot from an analysis session. The left frame shows coefficients of the linear models computed during the analysis. The dialog in the right frame is used to specify a linear regression model to compute. Integrated computation of the regression models makes the analysis faster and more efficient.

There are still many open research challenges in the field of ensemble analysis. As the data sets become more complex, ensembles from co-simulations will become common. Modern hardware-in-the-loop methods and resulting streaming data require novel solutions.

6 ACKNOWLEDGEMENTS

This work is partially written in scope of the COMET program at VRVis. VRVis is funded by BMVIT, BMWFW, Styria, SFG and Vienna Business Agency in the scope of COMET – Competence Centers for Excellent Technologies (854174) which is managed by FFG.

REFERENCES

Bergner, S., M. Sedlmair, T. Möller, S. N. Abdolyousefi, and A. Saad. 2013. "ParaGlide: Interactive Parameter Space Partitioning for Computer Simulations". *IEEE Transactions on Visualization and Computer Graphics* 19(9):1499–1512.

Boecking, F., U. Dohle, J. Hammer, and S. Kampmann. 2005. "Passenger Car Common Rail Systems for Future Emissions Standards". *MTZ worldwide* 66(7–8):552–557.

Boehner, W., and K. Hummel. 1997. "Common Rail Injection System for Commercial Diesel Vehicles". *SAE Transactions* (SAE 970345).

Cibulski, L., B. Klarin, M. Sopouch, B. Preim, H. Theisel, and K. Matković. 2017. "Super-Ensembler: Interactive Visual Analysis of Data Surface Sets". In *Proceedings of the 33rd Spring Conference on Computer Graphics*, SCCG '17, May 15th–17th, Mikulov, Czech Republic, article 19. ACM.

Demir, I., C. Dick, and R. Westermann. 2014. "Multi-Charts for Comparative 3D Ensemble Visualization". *IEEE Transactions on Visualization and Computer Graphics* 20(12):2694–2703.

Demir, I., J. Kehrer, and R. Westermann. 2016. "Screen-space Silhouettes for Visualizing Ensembles of 3D Isosurfaces". In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, April 19th–22nd, Taipei, Taiwan, 204–208.

Ferstl, F., K. Bürger, and R. Westermann. 2016. "Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 22(1):767–776.

Fröhler, B., T. Möller, and C. Heinzl. 2016. "GEMSe: Visualization-guided Exploration of Multi-Channel Segmentation Algorithms". *Computer Graphics Forum* 35(3):191–200.

Gresh, D. L., B. E. Rogowitz, R. L. Winslow, D. F. Scollan, and C. K. Yung. 2000. "WEAVE: A System for Visually Linking 3-D and Statistical Visualizations, Applied to Cardiac Simulation and Measurement Data". In *Proceedings of the IEEE Visualization 2000 Conference (VIS 2000)*, October 8th–13th, Salt Lake City, UT, USA, 489–492.

Haslett, J., R. Bradley, P. Craig, A. Unwin, and G. Wills. 1991. "Dynamic Graphics for Exploring Spatial Data with Application to Locating Global and Local Anomalies". *The American Statistician* 45(3):234–242.

Hauser, H. 2006. "Generalizing Focus+Context Visualization". In *Scientific Visualization: The Visual Extraction of Knowledge from Data*, edited by G.-P. Bonneau et al., 305–327. Berlin, Heidelberg: Springer.

Inselberg, A., and B. Dimsdale. 1987. "Parallel Coordinates for Visualizing Multi-dimensional Geometry". In *Computer Graphics 1987*, edited by T. L. Kunii, 25–44. Tokyo: Springer.

Kao, D., J. L. Dungan, and A. Pang. 2001. "Visualizing 2D Probability Distributions from EOS Satellite Image-derived Data Sets: A Case Study". In *Proceedings of the IEEE Visualization 2001 Conference (VIS '01)*, October 21st–26th, San Diego, USA, 457–460. IEEE.

Kehrer, J., P. Filzmoser, and H. Hauser. 2010. "Brushing Moments in Interactive Visual Analysis". *Computer Graphics Forum* 29(3):813–822.

Konyha, Z., K. Matković, D. Gračanin, M. Jelović, and H. Hauser. 2006. "Interactive Visual Analysis of Families of Function Graphs". *IEEE Transactions on Visualization and Computer Graphics* 12(6):1373–1385.

Lampe, O. D., and H. Hauser. 2011. "Model Building in Visualization Space". In *Proceedings of SIGRAD 2011. Evaluations of Graphics and Visualization – Efficiency; Usefulness; Accessibility; Usability*, November 17th–18th, Stockholm, Sweden, 43–51. Linköping University Electronic Press.

Luboschik, M., S. Rybacki, F. Haack, and H.-J. Schulz. 2014. "Special Section on Uncertainty and Parameter Space Analysis in Visualization: Supporting the Integrated Visual Analysis of Input Parameters and Simulation Trajectories". *Computer & Graphics* 39:37–47.

Matković, K., W. Freiler, D. Gračanin, and H. Hauser. 2008a. "ComVis: A Coordinated Multiple Views System for Prototyping New Visualization Technology". In *Proceedings of the 12th International Conference on Information Visualisation (IV '08)*, July 9th–11th, London, UK, 215–220.

Matković, K., D. Gračanin, M. Jelović, and H. Hauser. 2008b. "Interactive Visual Steering – Rapid Visual Prototyping of a Common Rail Injection System". *IEEE Transactions on Visualization and Computer Graphics* 14(6):1699–1706.

Matković, K., D. Gračanin, B. Klarin, and H. Hauser. 2009. "Interactive Visual Analysis of Complex Scientific Data as Families of Data Surfaces". *IEEE Transactions on Visualization and Computer Graphics* 15(6):1351–1358.

Matković, K., D. Gračanin, R. Splechtna, M. Jelović, B. Stehno, H. Hauser, and W. Purgathofer. 2014. "Visual Analytics for Complex Engineering Systems: Hybrid Visual Steering of Simulation Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 20(12):1803–1812.

Matković, K., D. Gračanin, M. Jelović, and H. Hauser. 2015. "Interactive Visual Analysis of Large Simulation Ensembles". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz et al., 517–528. Piscataway, New Jersey: IEEE.

Matković, K., H. Abraham, M. Jelović, and H. Hauser. 2017. "Quantitative Externalization of Visual Data Analysis Results Using Local Regression Models". In *Machine Learning and Knowledge Extraction*, edited by A. Holzinger et al., 199–218. Cham: Springer.

Montgomery, D. C. 2001. *Design and Analysis of Experiments*. 5th ed. New York: John Wiley & Sons.

North, C., and B. Shneiderman. 2000. "Snap-together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata". In *Proceedings of the Working Conference on Advanced Visual Interfaces*, May 24th–26th, Palermo, Italy, 128–135. ACM.

Piringer, H., S. Pajer, W. Berger, and H. Teichmann. 2012. "Comparative Visual Analysis of 2D Function Ensembles". *Computer Graphics Forum* 31(3pt3):1195–1204.

Potter, K., A. Wilson, P. Timo Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. 2009. "Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data". In *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, December 6th, Miami, USA, 233–240. IEEE.

Radoš, S., R. Splechtna, K. Matković, M. Đuras, E. Gröller, and H. Hauser. 2016. "Towards Quantitative Visual Analytics with Structured Brushing and Linked Statistics". *Computer Graphics Forum* 35(3):251–260.

Roberts, J. C. 2007. "State of the Art: Coordinated Multiple Views in Exploratory Visualization". In *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV '07)*, July 2nd, Zurich, Switzerland, 61–71. IEEE.

Sanyal, J., S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. J. Moorhead. 2010. "Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty". *IEEE Transactions on Visualization and Computer Graphics* 16(6):1421–1430.

Schulz, H.-J., A. M. Uhrmacher, and H. Schumann. 2011. "Visual Analytics for Stochastic Simulation in Cell Biology". In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies (i-KNOW '11)*, September 7th–9th, Graz, Austria, article 48. ACM.

Sedlmair, M., C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. 2014. "Visual Parameter Space Analysis: A Conceptual Framework". *IEEE Transactions on Visualization and Computer Graphics* 20(12):2161–2170.

Shrinivasan, Y. B., and J. J. van Wijk. 2009. "Supporting Exploration Awareness in Information Visualization". *IEEE Computer Graphics and Applications* 29(5):34–43.

Silva, C. T., J. Freire, and S. P. Callahan. 2007. "Provenance for Visualizations: Reproducibility and Beyond". *Computing in Science & Engineering* 9(5):82–89.

Sobol, I. 1976. "Uniformly Distributed Sequences with an Additional Uniform Property". *USSR Computational Mathematics and Mathematical Physics* 16(5):236 – 242.

Thomas, J. J., and K. A. Cook (eds.). 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. Los Alamitos, CA, USA: IEEE Computer Society.

Wang, J., X. Liu, H.-W. Shen, and G. Lin. 2017. "Multi-Resolution Climate Ensemble Parameter Analysis with Nested Parallel Coordinates Plots". *IEEE Transactions on Visualization and Computer Graphics* 23(1):81–90.

Whitaker, R. T., M. Mirzargar, and R. M. Kirby. 2013. "Contour Boxplots: A Method for Characterizing Uncertainty in Feature Sets from Simulation Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 19(12):2713–2722.

Wilson, A. T., and K. C. Potter. 2009. "Toward Visual Analysis of Ensemble Data Sets". In *Proceedings of the 2009 Workshop on Ultrascale Visualization*, November 16th, Portland, OR, USA, 48–53. ACM.

Yang, D., Z. Xie, E. A. Rundensteiner, and M. O. Ward. 2007. "Managing Discoveries in the Visual Analytics Process". *CM SIGKDD Explorations Newsletter* 9(2):22–29.

AUTHOR BIOGRAPHIES

KREŠIMIR MATKOVIĆ is a senior researcher at VRVis Research Center in Vienna. He is interested in extending visual analysis technology to challenging heterogeneous data, in particular to a combination of multi-variate data and more complex data types, such as functions. He also focuses his research on developing a structured model for visual analysis that supports a synergetic combination of user interaction and computational analysis. He teaches at TU Vienna where he received his doctoral degree and habilitation (in 1998 and 2015) and at the University of Zagreb, where he received his graduate degree in 1994. He is a member of ACM, Eurographics, and IEEE Computer Society. His email address is Matkovic@VRVis.at.

DENIS GRAČANIN received the BS and MS degrees in Electrical Engineering from the University of Zagreb, Croatia, in 1985 and 1988, respectively, and the MS and PhD degrees in Computer Science from the University of Louisiana at Lafayette in 1992 and 1994, respectively. He is an Associate Professor in the Department of Computer Science at Virginia Tech. His research interests include virtual reality and distributed simulation. He is a senior member of ACM and IEEE and a member of AAAI, APS, ASEE, and SIAM. His email address is gracanin@vt.edu.

HELWIG HAUSER is professor at the University of Bergen, Norway, where he is leading a research group on visualization since 2007. Before moving to Norway and since 2003, Helwig Hauser was the scientific director of the VRVis Research Center in Vienna, Austria. Earlier, he was assistant professor at the Vienna University of Technology, from which he also received his graduate and doctoral degrees (in 1994 and 1998) as well as his habilitation (2003). Helwig Hauser received several awards, including the biannual Heinz-Zemanek Award in computer science from OCG in 2006 and the Dirk Bartz Prize for visual computing in medicine in 2013. His email address is Helwig.Hauser@UiB.no.