

# Interactive Visual Steering of Hierarchical Simulation Ensembles

Rainer Splechtna\*

VRVis Research Center in Vienna, Austria

Krešimir Matković†

VRVis Research Center in Vienna, Austria

Denis Gračanin‡

Virginia Tech, Blacksburg, VA, USA

Mario Jelović§

AVL-AST Zagreb, Croatia

Helwig Hauser¶

University of Bergen, Norway

## ABSTRACT

Multi-level simulation models, i.e., models where different components are simulated using sub-models of varying levels of complexity, belong to the current state-of-the-art in simulation. The existing analysis practice for multi-level simulation results is to manually compare results from different levels of complexity, amounting to a very tedious and error-prone, trial-and-error exploration process. In this paper, we introduce hierarchical visual steering, a new approach to the exploration and design of complex systems. Hierarchical visual steering makes it possible to explore and analyze hierarchical simulation ensembles at different levels of complexity. At each level, we deal with a dynamic simulation ensemble — the ensemble grows during the exploration process. There is at least one such ensemble per simulation level, resulting in a collection of dynamic ensembles, analyzed simultaneously. The key challenge is to map the multi-dimensional parameter space of one ensemble to the multi-dimensional parameter space of another ensemble (from another level). In order to support the interactive visual analysis of such complex data we propose a novel approach to interactive and semi-automatic parameter space segmentation and comparison. The approach combines a novel interaction technique and automatic, computational methods — clustering, concave hull computation, and concave polygon overlapping — to support the analysts in the cross-ensemble parameter space mapping. In addition to the novel parameter space segmentation we also deploy coordinated multiple views with standard plots. We describe the abstract analysis tasks, identified during a case study, i.e., the design of a variable valve actuation system of a car engine. The study is conducted in cooperation with experts from the automotive industry. Very positive feedback indicates the usefulness and efficiency of the newly proposed approach.

**Keywords:** Interactive Visual Analysis, Simulation-Ensemble Steering, Multi-resolution simulation

## 1 INTRODUCTION

Strict emission regulations and increased environmental requirements are the main driving forces in modern automotive design. A modern car engine is a complex system that has many components. It exhibits behavior that is more complex than a simple combination of the behaviors of the individual components. Engineers use simulation models at various levels of detail when simulating engines. Due to prohibitively long simulation times it is not feasible to simulate the whole engine at the finest level of detail.

The current state of the art in simulation is to represent individual components with simplified models when studying the whole

system, and using more detailed models of individual components when they themselves are in the design focus. During the design process, numerous simulation runs are computed using models of varying complexity. It is essential that coarse simulation models represent the designed system sufficiently well. Although visualization and visual analysis are extensively used to study results from single-model simulations, there are practically no tools to support the analysis of multi-resolution simulations. The standard approach is to manually compare simulation results from different simulation models, resulting in a tedious exploration of a very large simulation model space.

In this paper we study the challenge of complex system design. We build on results from studying a single-level simulation model [11], extending the work to now manage multi-resolution simulation models. The new approach, called *Hierarchical Visual Steering*, is a result of a long-time collaboration of visualization, optimization, and engineering experts. Finding such a solution is only possible with an interdisciplinary team. In contrast to conventional visualization, where data does not change throughout the analysis, we deal with ensemble steering — new simulation runs are started from the visualization and the data changes during the analysis session. Furthermore, and since we deal with multi-resolution simulation models, we need to handle several, dynamically growing simulation ensembles simultaneously.

The key challenge is to map the multi-dimensional parameter space of one ensemble to the only partially overlapping multi-dimensional parameter space of another ensemble (at another level). Due to the system model's complexity and the lack of an analytical description of the whole system, the analytical mapping of parameter spaces is practically impossible. We propose a novel, interactive approach to solve this problem. We introduce an interactive, semi-automatic parameter space segmentation approach. It starts with a concave hull computation of the brushed points, and continues (either automatically or user-triggered) by employing clustering in order to find more compact regions. Once compact areas with desired behaviors are found, they can be compared. All of this happens at the individual model level. We also introduce an output-based cross-level parameter mapping approach. By means of this mapping the compact areas of desired outputs behaviors from one level can be compared with the areas from other levels. This makes it possible to check for the physical plausibility of coarse models, before using them in the larger models.

Our solution enables the validation of the design and reveals insights that can be used to develop and refine more general visualization guidelines (patterns). The domain experts were involved from the beginning of this research, allowing us also to leverage participatory design in order to identify the tasks and requirements of the automotive design process. The real-world problem addressed here is the design of a variable valve actuation (VVA) system, i.e., a complex system that controls the opening and closing of the intake and the exhaust valves of a four stroke combustion engine. It increases engine efficiency, decreases consumption, and reduces the overall emissions.

The main contribution of this paper is *a novel approach to the interactive visual analysis and design of real-world complex sys-*

\*e-mail: Splechtna@VRVis.at

†e-mail: Matkovic@VRVis.at

‡e-mail: gracanin@vt.edu

§e-mail: mario.jelovic@avl.com

¶e-mail: Helwig.Hauser@uib.no

tems – hierarchical visual steering. Additional contributions, necessary to realize the hierarchical visual steering, include: **1)** Semi-automatic and interactive parameter space segmentation. **2)** Output-based cross-level parameter space mapping. **3)** Analysis procedures and tasks, abstracted for the analysis of complex, multi-resolution simulation results.

## 2 RELATED WORK

Simulation steering and exploring ensemble simulations require multi-resolution simulation models [4] and good control over multiple heterogeneous simulation runs. World lines [20] integrate simulation, visualization and computational steering to deal with the extended solution space by representing simulation runs as causally connected tracks, sharing a common time axis. World lines do not support multi-resolution simulation models, in contrast to our approach. The studied model does not change during the analysis.

In each computational steering iteration the user defines a region of interest to be further explored. Additional simulation runs, which are needed to cover this region of interest, are called a new “simulation experiment” [10]. The design of such an experiment, i.e., the selection of the simulation points in the region of interest, is very important since we would like to keep the number of simulation runs low, while still providing a good coverage of the region of interest [7].

Sedlmair et al. [16] describe a conceptual framework for visual parameter space analysis. The framework encompasses a data flow model, four parameter space navigation strategies and six analysis tasks. They also provide an overview of the related visual parameter space analysis literature and identify three research gaps — the data acquisition gap, the data analysis gap, and the cognition gap.

The approach of a user-controlled simulation is at the very core of computational steering – still, only few corresponding solutions exist. Very often it is impossible to acquire an optimal solution. The user has to interactively analyze trade-offs and interdependencies between different objectives [17, 19]. Using an analytical representation of the objective function, the values of the objective function in the region of interest can be presented to the user [12]. These techniques guide the user to potentially interesting parameter regions, while showing the uncertainty of predictions.

Piringer et al. [13] describe an interactive approach, HyperMoVal, designed to support multiple tasks related to model validation. Bergner et al. [2] introduce ParaGlide, a visualization system for the interactive exploration of parameter spaces of multidimensional simulation models. They also support parameter space segmentation. They rely on the clustering of the output space and depict the corresponding points in different colors in the control parameters views. We propose a novel, semi-automatic parameter space segmentation that uses clustering and iterative refinement. In contrast to our approach, all approaches mentioned above only deal with a single ensemble.

The VVA (variable valve actuation) problem domain deals with opportunities to decrease fuel consumption in modern combustion engines [14, 15]. Lancefield et al. [9] provide an overview of the design solution and a description of the simulation model used for VVA systems. Denger and Mischer [5, 6] describe an electrohydraulic valvetrain system, a subset of variable valve actuation systems. Interested readers are referred to above mentioned papers as a starting point in VVA research. A broader description of the VVA design would go beyond the scope of this paper.

## 3 TASK ABSTRACTION

The overall design of a complex system, such as a car engine, starts with a detailed design of its components. Several design teams optimize different components to create a coarse simulation model. Groups of components (subsystems) are analyzed based on the coarse simulation model used in a simulation process.

The simulation process can be represented as a function  $S$  that maps the (control) parameters  $\mathbf{x} = (x_1, \dots, x_m)$  to the output values  $\mathbf{y} = (y_1, \dots, y_n)$  where  $m$  is the number of parameters and  $n$  is the number of output values. A set of pairs  $(\mathbf{x}, \mathbf{y})$ ,  $\mathbf{x} \in C$  (parameter space) and  $\mathbf{y} \in O$  (output values space) is a simulation ensemble  $E$ .

During the engine design phase, the coarse subsystems are simulated together and the whole system is explored, analyzed, and tuned. The selected system designs are validated using more detailed simulation models. The main principle is always the same: the user wants to find the coarsest model possible that is still physically plausible. A coarse model is a rough approximation, its physical plausibility has to be checked using finer models. At the very end of this process, the production can start.

We have observed the basic tasks (and the corresponding basic patterns) that are performed repeatedly during the analysis and verified the relevance of these tasks with colleagues from other design groups (timing drive, driving comfort, vibration reduction, etc.). In this paper, we focus on tasks that are specific to multi-resolution simulation, but typical tasks for single ensemble steering [11] are also applicable to hierarchical steering. The steering process between two levels of complexity (pair-wise ensemble comparisons) is shown in Figure 1.

The tasks common to single ensemble steering (TS) include:

- **(TS1) Output Analysis:** Find combinations of control parameters that produce desired (or undesired) output values.
- **(TS2) Model Reconstruction:** Analyze how the outputs change as the values of control parameters change.
- **(TS3) Ensemble Growing:** Add simulation runs with new control parameters to the currently analyzed ensemble.

The tasks specific to hierarchical steering (TH), being in the particular focus of this paper, can be summarized as follows:

- **(TH1) Model Refinement:** Create an ensemble for a (partially refined) model. Add a new ensemble with a different control parameter space to the set of ensembles. The ensemble space is growing now, in contrast to the task **TS3** where one ensemble itself grows.

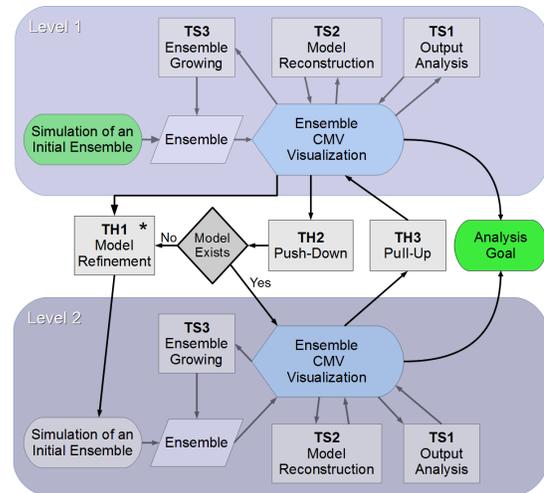


Figure 1: Starting from an initial ensemble for the lower level of complexity (Level 1), the analysis/steering process executes a combination of basic steering tasks to create more complex models (Level 2) and, consequently, ensembles until the analysis goal is reached, i.e., subsets of the analyzed ensembles which satisfy this goal are found.

- **(TH2) Push-Down:** The currently identified region of interest in the parameter space of an ensemble is mapped – *pushed down* – to an ensemble of a finer resolution. This is realized with a link between a coarse and a fine ensemble (multiple links are possible). This cross-model-link enables the exploration of the dependencies and correlations between two levels of the model space, using “forward-mapping”. With this pattern, it is possible to analyze how the shapes of corresponding regions in one parameter space change as the values of the control parameters in another ensemble change.
- **(TH3) Pull-Up:** The current state of the analysis in a fine-level ensemble is mapped – *pulled up* – to the coarse-level ensemble by establishing a link between the two ensembles (multiple links are possible). This cross-model-link enables the exploration of the dependencies and correlations between two levels, using “backward-mapping”. It is possible, e.g., to analyze which combinations of control parameters in the coarse ensemble produce outputs that are similar to the desired outputs in a finer ensemble. If two ensembles share some control parameters, we compare the combinations of these control parameters that produce similar outputs on both levels. Using the established links between all existing fine and coarse ensembles we analyze the whole model space concurrently.

In order to support the above identified tasks we propose a novel interactive solution for parameter space segmentation and output-based parameter space mapping, as described in the following.

#### 4 SEMI-AUTOMATIC, INTERACTIVE, PARAMETER SPACE SEGMENTATION

The task **TS1**, Model Reconstruction, examines which parameter settings correspond to a desired simulation output. After an initial simulation ensemble is computed, the exploration starts. Coordinated, multiple views are used with at least one view showing parameters and at least one other view showing output values. Desired output values are brushed, consequently the points in the parameter space views are highlighted. The corresponding parameters are studied. Regions in the parameter space that result in certain desired outputs are in the main focus of the exploration. Figures 2a and 2b illustrate this phase of the process. Curves of a desired shape are selected. The selection is additionally refined in other views (not shown in the Figure 2 due to the space constraints). The brushed curves are shown in dark green, and the corresponding control parameters are shown in the scatterplot in Figure 2b. Note that non-brushed points (shown in gray) are interwoven with brushed points.

The interactive segmentation process starts. A concave hull using the brushed points as input is computed. The concave hull is used since the more usual convex hull increases the hull area significantly in the case where outliers are present. The implemented concave hull algorithm allows the user to set a parameter that controls the smoothness of the concave hull. In most of the cases the hull will contain non-brushed points as well. Figure 2c shows the computed concave hull. Light green points in the concave hull correspond to the non-brushed points within the hull. In order to quantify results we depict the number of brushed points and the total number of points in the hull in the segmentation control dialog (Figure 2d).

Depending on the number of non-brushed points in the hull the user can proceed with different actions. Usually the first action is to check the non-brushed points. To do so, the user just selects the check option from the segmentation control dialog and the points are implicitly brushed. The corresponding outputs are depicted in a different color. Figure 2e shows the checked curves in light green. Note that some of them have unwanted shapes, so we cannot keep the whole hull as a selection. In this case the user selects to split the hull into several smaller hulls (clusters) instead of accepting the

hull. The user decides on the number of clusters and the clustering algorithm creates these new clusters of the brushed points of the hull. Each cluster is contained in the original hull. Figure 2f shows three clusters. The control dialog shows the hierarchy of the clusters, information for each cluster, and provides controls to further check or split any of the new hulls.

In general, the new concave hulls will contain fewer non-brushed points than the original concave hull. The user can check each hull, accept the hull, or split it further. Some regions, with too many non-brushed points, will be rejected and not taken into further consideration. In our case, the clusters A and C are acceptable (Figure 2g shows the checked curves for cluster A). In contrast, cluster B results in some curves with undesired shapes (Figure 2h), so the cluster is split further (Figure 2i).

Eventually, the user identifies homogeneous regions in the parameter space corresponding to the desired outputs (if such regions exist), and can accept the corresponding hulls as useful parameter selections, assign them a name/label (such as “Steep Rise”), and a color.

If the number of points in a region drops below a certain threshold, the user can request new simulation runs directly in the visualization (**TS3**, Ensemble Growing). This highly interactive and iterative process supports the parameter space segmentation. Figure 3 illustrates the whole process. At the end, all regions from the analysis are added to the regions list which is stored globally. During the process the regions list will grow. There will be regions, e.g., corresponding to a low engine consumption, high power, and low emission. Each of them gets its color and can be depicted if desired. A cross section of selected regions can also be computed and displayed.

It is always possible to automatize the whole process to proceed automatically with splitting if there are too many non-brushed points in a region. The user still has to cross-check regions and approve the final regions before they are accepted. We have used the k-means clustering that worked well since the user interventions were minimal. Other clustering algorithms can be included and selected by the user.

#### 5 OUTPUT-BASED, CROSS-LEVEL PARAMETER SPACE MAPPING

After the segment action of the parameter space at different levels, it is necessary to map segments corresponding to the same behavior (low consumption, high power, ...) between the levels. As the parameter spaces are different, i.e., the parameters overlap only partially or not at all, and there is no explicit mathematical formula describing the entire system under investigation, an analytical mapping of parameter values is not possible. Hence we propose an output-based, cross-level mapping between parameter spaces at different levels of complexity. The main idea is to compare the outputs from different levels of complexity, and to identify parts of the parameters spaces where these outputs are similar (according to a user defined metric). The user selects a segment in the parameter space of the more complex simulation model (ensemble  $E_2$ ), which should be pulled up (**TH3**) to a less complex simulation model (ensemble  $E_1$ ). A parameter space region  $R_{E_2}$  corresponds to a subset of output values in  $E_2$ . Now, the set of output values is compared with the corresponding set of output values in  $E_1$ .

Similar output values are selected, and an implicit, cross-level brush is created. The similar output values are brushed in  $E_1$ , and the corresponding points in the parameter space are shown. The parameter space can be segmented again, and cross-sections with the original segments are found. Overlapping regions indicate corresponding areas in the parameter spaces of different levels.

A proper metric is needed to compare output values. The control parameters are usually numerical, scalar values. Often, however, the output values also include time series data so that the output

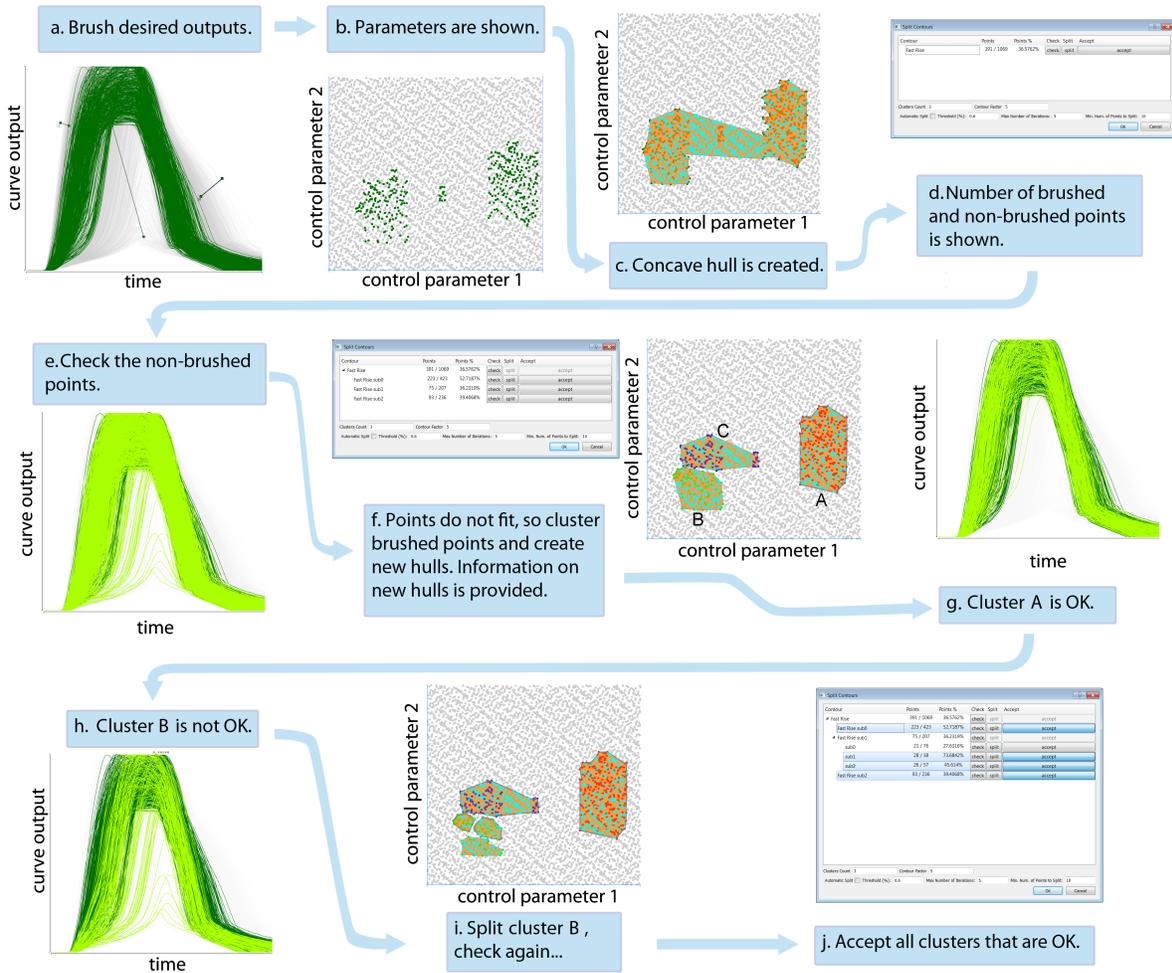


Figure 2: A typical workflow of interactive, semi-automatic segmentation. The user goes through steps a. to j. Almost every step can be refined, the user can go back and forth at any time. The figure illustrates the main steps without such refinement steps.

space  $O$  is no longer a subset of  $\mathbb{R}^n$ . Since simple data tables [3] are not sufficient, we use a two-level data model for the output data points (Figure 4). Every output data point  $\mathbf{y}^i$  and  $y_j^i$  that is a data series has a separate set of “sub-points” with its own length and number of dimensions (we limit our discussion to two dimensions). Such a data series is viewed as a function of one variable (time) and represented as a curve. Hence, we have to define an appropriate metric for both kind of output values, scalars and time series.

When dealing with scalar outputs we allow the user to set a tolerance  $\epsilon_u$  for each scalar output dimension  $u$  that should be compared. All output values in  $E1$  that are within the tolerance range, as compared to the corresponding values in  $E2$ , are selected. We can combine several outputs using Boolean operators. We can describe the cross-level, output-based mapping for matching scalar attributes  $u1$  in ensemble  $E1$  and  $u2$  in ensemble  $E2$  as follows:

$$TH3(R_{E2}) = \{\mathbf{y}^i \mid \mathbf{y}^i \in E1 \wedge \exists \mathbf{y}^j \in R_{E2} : \|y_{u1}^i - y_{u2}^j\| \leq \epsilon_u\} \quad (1)$$

The metric is a bit more complicated for time series outputs. The first idea was to use the root mean square error (RMSE) method. This simple similarity measure does not seem to be powerful enough to capture the various ways in which output curves can be similar/dissimilar to each other. Figure 5 top depicts output

curves from two ensembles for the same output value, i.e., the valve position, described in more detail in Section 6. The coarse model (Figure 5 left) cannot simulate fine details such as the smooth valve closing that is visible in the fine model curves (Figure 5 right). The slow decrease to zero is a desired behavior which simply cannot be simulated using a coarse model. If we would compute RMSE differences, the errors would be large. Instead, we allow the user to interactively specify a range of comparison  $r_v$  for matching time series dimensions and the corresponding tolerance  $\epsilon_v$ . Equation 2 describes the output-based mapping for matching time series attributes  $v1$  in ensemble  $E1$  and  $v2$  in ensemble  $E2$ .

$$TH3(R_{E2}) = \{\mathbf{y}^i \mid \mathbf{y}^i \in E1 \wedge \exists \mathbf{y}^j \in R_{E2} : RMSE(y_{v1}^i, y_{v2}^j, r_v) \leq \epsilon_v\} \quad (2)$$

$RMSE(y_{v1}^i, y_{v2}^j, r_v)$  is applied to curves  $y_{v1}^i$  from  $E1$  and  $y_{v2}^j$  from  $E2$  within the range  $r_v$ . If we limit the range correctly, the RMSE will select similar curves. Figure 5 bottom row shows similar curves after range was limited. The user selected the desired shape in the coarse model (Figure 5 bottom left), and similar curves in the fine model are found (Figure 5 bottom right).

We can specify a TH3 mapping also as a combination of several scalar and time series outputs. Equations 1 and 2 are applied to

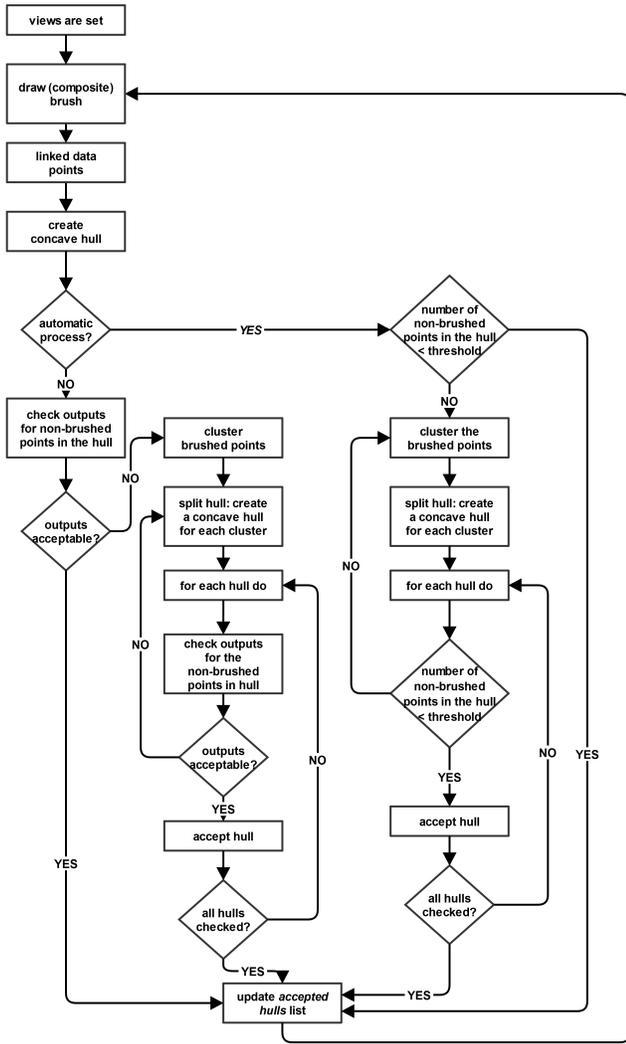


Figure 3: Flow chart of the interactive, semi-automatic parameter space segmentation process. An automatic mode, based on the relative number of non-brushed points, and an interactive mode are supported.

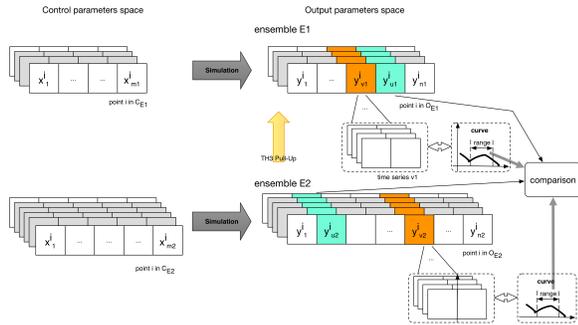


Figure 4: Simulation ensemble data model: control data points, output data points and comparison. Two ensembles,  $E1$  and  $E2$ , have only partially overlapping parameter spaces. We select a region, i.e., collection of output data points (orange), in the corresponding output spaces and compare the scalar values (cyan).

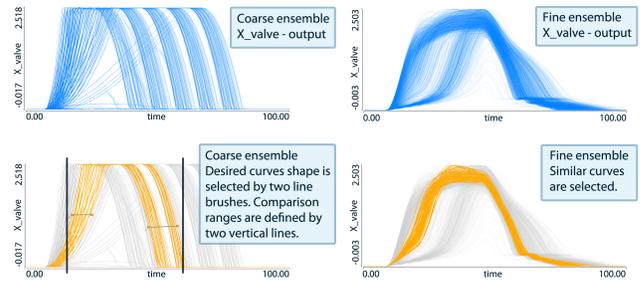


Figure 5: An example of the output-based mapping principle. The same output curves from two different ensembles have different shapes. The user decides which output dimension will be mapped ( $X_{valve}$  in this case), and, optionally limits the x-axis range for comparison. The RMSE is computed and all curves that are within a threshold (10% here) are selected in the other ensemble.

individual outputs and the resulting sets of data points in the output space of ensemble  $E1$  are combined using basic set operations.

This metric proved to be useful for our case study. In some other domains, and with different data, users will probably need different metrics, but the main principle of the output-based mapping remains the same. The desired outputs from one level (ensemble) are compared to the same outputs in another level (ensemble) using an appropriate metric. The similar outputs are then selected and the corresponding parameters are explored. In this way, regions from different parameter spaces can be mapped to each other and thus compared.

## 6 CASE STUDY: VARIABLE VALVE ACTUATION SYSTEM

We use a car engine design case study to illustrate the proposed hierarchical ensemble steering. The case study was conducted on a single workstation (including simulation). Variable valve actuation (VVA) is an active research field in the automotive industry and it is closely related to the development of four-stroke engines. A precise control of the opening and the closing of the intake and the exhaust valves is essential for an optimal engine operation. 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 cam-less system. Such a flexible system can ensure the variable feeding and dissipation of the gases involved in the combustion process.

Our goal is to understand the system behavior, to set system parameters (valve opening curves), and to create a coarse, yet plausible, simulation model which will be integrated with other models on a larger scale. Figure 6a. shows the simulation models, at different levels of complexity (case study). There are three blocks in our complex system, the control block, the actuator block, and the valve block. Two are refined during the analysis (one at a time), resulting in three different simulation models, and three ensembles, accordingly. A separate color is assigned to each model in order to identify the related ensemble/parameters/results during the hierarchical visual steering process.

Once the coarsest model is tuned, we refine a component of our choice and start the whole process over. We focus on three simulation results in our study: the hydraulic oil quantity that enters the system ( $Inflow$ ), the pressure in the actuator block ( $P_{act}$ ), and the valve position ( $X_{valve}$ ). All three variables (outputs) are time series data, i.e., functions of time, for which we also compute selected scalar features. We have a complex data set, consisting of many data points (each corresponding to one simulation run), and many attributes per data point. The attributes are scalar values and functions of time [8].

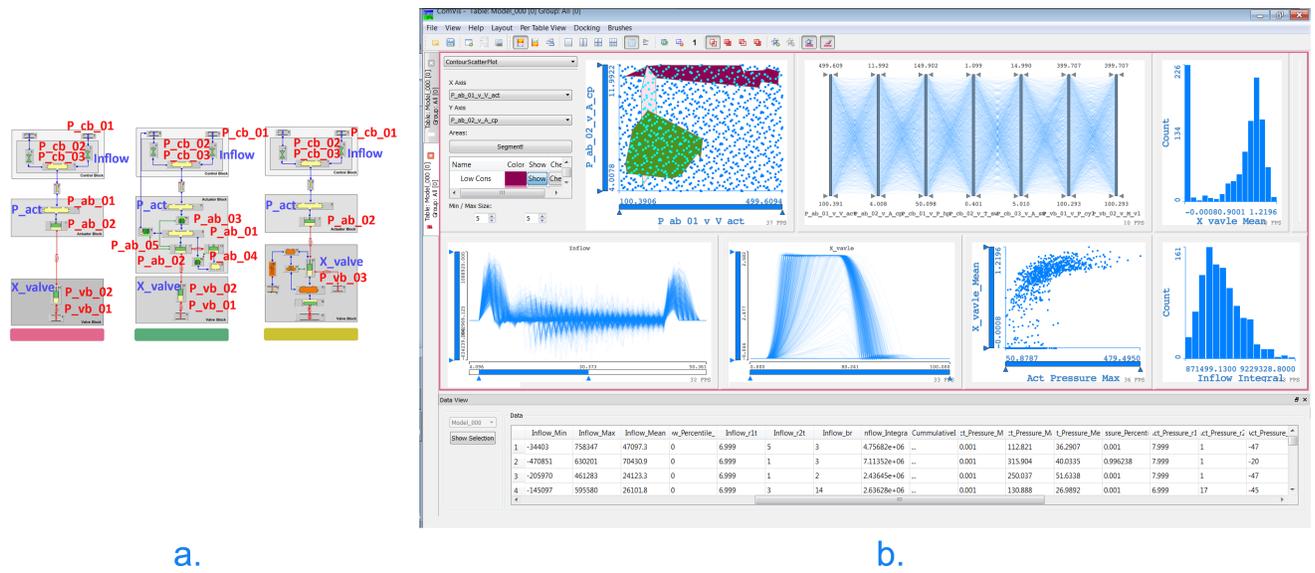


Figure 6: **a.** The initial, low resolution model of the VVA system (left) and the two refined models were designed with AVL Boost-Hydsim [1]. Each model and the corresponding ensemble has an assigned color (bar underneath). The control parameters under investigation in our case study are depicted in red, and three main state variables are shown in blue. **b.** A snapshot from a single-level analysis. The newly introduced interactive segmentation view is shown in top left. Different other views show various control parameters and output values of the ensemble. The data table in the bottom provides details on demand for selected runs.

Table 1: Initial Parameter Ranges for the First Two Levels

Parameter	Coarse Model	Fine Model
Actuator volume size $P_{ab.01.v.v.act}$	100–500 mm <sup>3</sup>	
Actuator piston area $P_{ab.02.v.A.cp}$	4–12 mm	8–12 mm
Inflow pressure $P_{cb.01.v.P.hp}$	50–150 bar	80–120 bar
Opening/Closing time $P_{cb.02.v.T.sw}$	0.4–1.3 ms	0.4–1.2 ms
Max. flow area $P_{cb.03.v.A.sw}$	5–15 mm <sup>2</sup>	
Cylinder pressure $P_{vb.01.v.P.cyl}$	100–500 N	100–400 N
Valve mass $P_{vb.02.v.M.v}$	100–400 g	
Port cut discharge coeff. $P_{ab.03.v.My.cut}$		0.6–1.0
Damper discharge coeff. $P_{ab.03.v.My.damp}$		0.7–1.0

The first step is to identify relevant parameters which may be varied (Figure 6a. shows the parameters that were varied for each model in red). We use a Sobol sequence [18] to vary the parameters in order to cover the whole multi-dimensional parameter space appropriately. We vary seven parameters in total. We start our analysis of the coarse model with 1,000 runs. During the analysis, the initial ensemble grows to 2,603 runs. In the next step, we create a fine initial ensemble consisting of 2,000 runs, which then also grows during the analysis. Table 1 shows the initial parameter values for the first two models.

### 6.1 Single-Level Analysis

At one particular level we support three principal tasks. We select desired curve shapes (in the VVA system, e.g., the valves have to open and close as fast as possible), and check the corresponding values of the control parameters (**TS1**). If the control parameter space is not sufficiently sampled we can run new simulations (**TS3**). Additionally, we also often check what happens with the simulation results as we change the control parameters (**TS2**).

Figure 6b. shows the coarse-level ensemble analysis. The already identified regions are shown in the top left view. There is a list of all identified regions and the user can show or hide each of them

separately. The parallel coordinates plot shows the control parameters. Two histograms show two scalar features of the time series outputs. Two curve views at the bottom show the time-dependent simulation results of the current ensemble— *Inflow*, and *X\_valve*. The scatterplot shows two additional features. Each simulation run is represented with one poly line in the parallel coordinates plot, one point in the scatterplot, and one curve in each curve view. As all views are linked, we can select items in each view and see the corresponding values in all other views. The user simply selects a desired range on one of the parallel coordinates, or draws a rectangle in the scatterplot, selects bins in the histogram, or uses a line brush [8] in a curve view. The scatterplots of the control parameters support the above described semi automatic segmentation, and several regions have already been identified.

The coarse-level analysis shows that the area of the control piston, the pressure in the cylinder, and the valve mass, apparently influence the closing speed substantially. The inflow pressure at the high-pressure valve and the ratio of opening and closing time at high pressure valve influence the closing speed much less, and the size of the actuator volume and the maximum flow area at the high pressure valve have almost no influence. After we have tuned the initial, unrefined model, a finer ensemble is defined and simulated (**TH1**). The finer ensemble is analyzed in a similar way.

An often observed pattern in a single-level analysis is to start with constraining control parameters to some desired values. Figure 7 shows such an example. For a fine-level ensemble, we brushed desired curve ranges and the corresponding output values are shown. The selections are refined as long as the output values are not desired. Once the desired outputs are found we have a segmentation of the control parameters space, and we can proceed with a cross-level analysis.

So far we have described the separate exploration of one particular ensemble. Studying the different ensembles separately often results in an unnecessarily large number of trials (and errors). Both models represent the same physical system, at different levels of detail — their parameter spaces are also different. The main challenge now is how to compare the parameter spaces of different models,

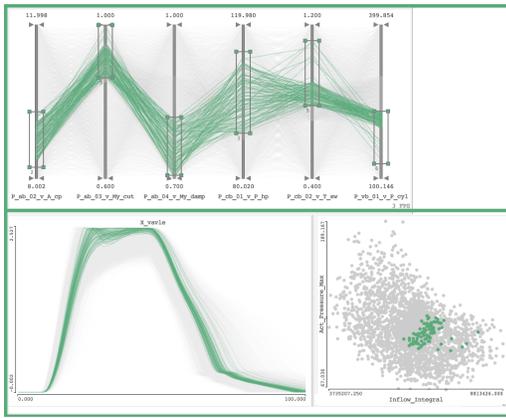


Figure 7: The 2,000 initial runs for a fine level are computed and we select the preferred control parameter ranges. The  $X_{valve}$  curves are shown in the curve view and the scatterplot shows the integral of the Inflow on the x-axis, and the maximum actuator pressure on the y-axis.

and how to map one parameter space to the other.

## 6.2 Cross-level Analysis

At each level we identify interesting regions in the parameter space. In order to compare the regions we use the above described output-based mapping. In our case, we use the  $X_{valve}$  curves that are available in both of our ensembles. They do have different shapes, but they both represent the valve position. Therefore, we establish the first cross-level link using these valve position curves.

For the main cross-level analysis (TH2 and TH3) we need a cross-level link and a specially designed implicit brush to support a cross-level parameter space exploration. The cross-level analysis works the same for two or more levels. The main idea is to identify curves of desired shape and the corresponding parameters in one ensemble first. Then, we find similar curves in the other ensemble. These curves are brushed implicitly, i.e., by means of their similarity as compared to the desired curves — the corresponding runs are highlighted accordingly. Figure 8 illustrates this process. Figure 8a shows the control parameters and the valve position curves from the coarse ensemble. Figure 8b shows the parameters and curves for the fine ensemble. We interactively drill down to two curves selected in the fine ensemble (Figure 8c). Now the established cross-link finds curves in the coarse ensemble which are similar to the two brushed curves from the fine ensemble. Based on defined similarity metric there are 42 similar curves. They are implicitly brushed in the coarse ensemble using the fine ensemble color (Figure 8d). In this way we know that this selection originates from another level. The parallel coordinates in Figure 8d show the control parameter values for the coarse model which result in curve shapes that are similar to the desired curves in the fine model. The analysis continues by selecting within 42 runs the setup for control parameters using additional criteria or domain experts preferences. The model accuracy can be determined by checking overlapping of the parameter space regions.

## 7 USER FEEDBACK AND EVALUATION

The interactive hierarchical ensemble steering has been developed in close collaboration with domain experts. We have observed their workflow, when designing complex systems, and realized how tedious the whole process is. The improvements presented in this paper are results of a common endeavor during a one-year project. The engineers, who were involved in this project, are experienced

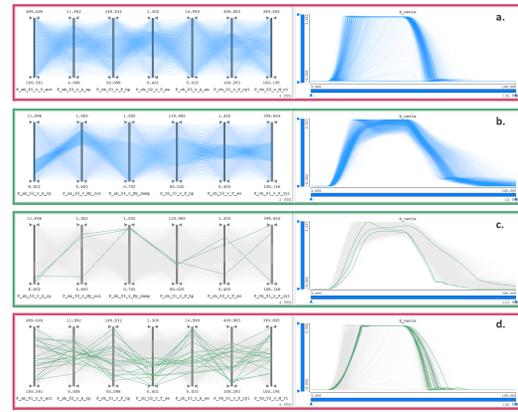


Figure 8: Cross-level link illustration. **a)** A coarse ensemble, initial parameters (Table 1), in the parallel coordinates plot and the corresponding  $X_{valve}$  curves. **b)** The corresponding views in a fine ensemble. **c)** Three curves selected in the fine ensemble. The cross-level link is activated, the range and threshold for the comparison are selected. **d)** The corresponding curves in the coarse ensemble.

with interactive visual analysis and with interactive steering. However, they did not have any support tools for any cross-level analysis. The standard workflow includes numerous manual comparisons and requires many trials until a satisfactory design is found.

We asked three domain experts to estimate the speedup when using the new approach. They indicated a speed-up of an order of magnitude (under an hour versus the whole day) using the hierarchical steering over the standard workflow (manual comparisons). The estimation from the case study confirms this. The domain experts also emphasized that, in addition to speed-up, the new approach also provides better insight. When ensembles are compared manually, the domain experts always suspected that there is something they did not see.

The semi-automatic parameter space segmentation using scatterplots is very intuitive and appreciated, but it shows only one possible 2D projection of the multidimensional parameter space. The engineers were thus using several scatterplots in parallel. Sometimes, clear regions appear in some projections, but in others they were comparably disconnected. Also, we were forced several times to modify the simulation models. Although there were parameter combinations that resulted in desired behavior, they still were considered to be very unstable. All points in their neighborhood had undesired behavior, and it was simply too risky to continue with such a sensitive system.

We are currently researching how to improve the support for multidimensional parameter space segmentation. One of the engineers stated that *“This is, no doubt, a tremendous step forward compared to the current state of the art, but it also made us aware of many challenges that are yet to be solved.”*

As the number of parameters grow, 2D projection becomes less and less useful. At the same time, as the system complexity grows, it becomes practically impossible to comprehend the overall system behavior without the support of visual tools. During the evaluation we have also created a toy example, consisting of two springs and two bodies that only can move along a line. Although the analytical solution of the problem is clear, engineers had a problem to mentally comprehend the overall system behavior as masses and the stiffness of the springs change. While experimenting with this simple model, one of the experienced engineers stated that *“In order to understand the behavior of a complex, non-linear systems, we do need a strong mathematical support to understand the fundamental laws, but the addition of interactive visualization enables*

us to answer some complex questions that are relevant to our engineering practice. Only the combination of both can help us with really complex tasks.”

One of the engineers (who also coauthors the paper) who is familiar with interactive visual analysis was very enthusiastic about using this approach for the variable valve design. He stated that “Automatic clustering as a basis for splitting and the corresponding check feature are the most useful segmentation support tools I’ve used up to now.” While all case studies were in automotive design, the positive feedback from the engineers indicates the applicability to other domains.

## 8 DISCUSSION AND CONCLUSION

We have described a new approach to interactive visual steering, i.e. *hierarchical visual steering*, that operates on a multi-level simulation model space. This approach bridges the often diverging parameter spaces of models at different levels of complexity and enables the exploration of ensembles from different levels without losing the link to the overall model. Such support tools are essential for complex systems design. As the system complexity grows, we simply need visual tools in order to comprehend the system behavior. This is especially true for non-linear systems.

We identified main tasks in the hierarchical steering workflow and proposed a first solution to the problem. The proposed solution scales well with the number of data points, and with the number of ensembles. System designers often use models of several levels of complexity, but a pairwise comparison, if applied successively, can cope also with a large number of ensembles.

The multi-dimensionality of the parameter space is a difficult challenge that we are addressing with our approach. We use 2D projections of the parameter space. As the number of parameters grows, the number of possible 2D projections grows even faster. The current state of the art in the parameter space exploration simply cannot cope with such high-dimensional spaces, unless their dimensionality can be somehow reduced prior to the analysis. Some authors, fix all parameter values and analyze what happens when changing only one (or two) parameters at a time. We see a great potential for future research here.

The feedback from the domain experts confirms the usability of the newly proposed analysis approach. For the first time, according to our best knowledge, we support the simultaneous exploration of multiple, dynamic ensembles. The interactive visual analysis across ensembles from different models is supported by the cross-model link, so the consequences of refining can be interactively explored by comparing the outputs/results/findings and ranges of optimal control parameters across different models in the model space. The heterogeneous parameter spaces are linked, so for each (directly) brushed parameter combination in one ensemble, the corresponding simulation runs from the other ensembles are indirectly selected. The new interactive, semi-automatic segmentation, as well as the newly proposed output-based, cross-ensemble mapping support the proposed approach.

When using conventional approaches, the complex system comprehension is very difficult due to its sheer complexity. It became possible for us to optimize the entire VVA system with a very large number of parameters due to this hierarchical divide-and-conquer strategy. The ability to optimize individual blocks within a large system, while still keeping the overall view onto the simulation up-to-date, proves to be very efficient and effective. Although the approach was developed in conjunction with this VVA system design, the very positive feedback from the involved domain experts makes us confident that this new approach can be applied to other complex systems as well. Future work will include a more formal evaluation of the newly proposed approach, especially when dealing with higher-dimensional parameter spaces.

## REFERENCES

- [1] AVL. AVL List GmbH. <http://www.avl.com/>. [last accessed July '15].
- [2] S. Bergner, M. Sedlmair, T. Möller, S. N. Abdolyousefi, and A. Saad. ParaGlide: Interactive parameter space partitioning for computer simulations. *IEEE TVCG*, 19(9):1499–1512, Sept. 2013.
- [3] S. K. Card, J. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Interactive Technologies. Morgan Kaufmann, 1999.
- [4] J. C. Carnahan, P. F. Reynolds, Jr., and D. C. Brogan. Visualizing coercible simulations. In *Proceedings of the 36th conference on Winter simulation*, pages 411–419. Winter Simulation Conference, 2004.
- [5] D. Denger and K. Mischker. The electro-hydraulic valvetrain system. *MTZ worldwide*, 65(12):10–13, 2004.
- [6] D. Denger and K. Mischker. The electro-hydraulic valvetrain system ehvs - system and potential. In *SAE Technical Paper 2005-01-0774*, SAE World Congress, Detroit, MI, USA, 2005.
- [7] J. P. C. Kleijnen. *Design and Analysis of Simulation Experiments*. International Series in Operations Research & Management Science. Springer, New York, 2007.
- [8] Z. Konyha, K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive visual analysis of families of function graphs. *IEEE TVCG*, 12(6):1373–1385, Nov./Dec. 2006.
- [9] T. Lancefield, I. Methley, U. Räse, and T. Kuhn. The application of variable event valve timing to a modern diesel engine. Technical Report 2000-01-1229, Mechadyne International Ltd, 2000.
- [10] K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive visual steering — rapid visual prototyping of a common rail injection system. *IEEE TVCG*, 14(6):1699–1706, Nov.-Dec. 2008.
- [11] K. Matković, D. Gračanin, R. Splechna, M. Jelović, B. Stehno, H. Hauser, and W. Purgathofer. Visual analytics for complex engineering systems: Hybrid visual steering of simulation ensembles. *IEEE TVCG*, 20(12):1803–1812, Dec. 2014.
- [12] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3):909–922, May 2006.
- [13] H. Piringer, W. Berger, and J. Krasser. HyperMoVal: Interactive visual validation of regression models for real-time simulation. *Computer Graphics Forum*, 29(3):983–992, June 2010.
- [14] J. Pohl, A. Warell, P. Krus, and J.-O. Palmberg. Conceptual design of a hydraulic valve train system. *Acta Polytechnica*, 4(4–5):20–28, 2001.
- [15] 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.
- [16] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE TVCG*, 20(12):2161–2170, Dec. 2014.
- [17] J. Seo, M. Bakay, Y.-W. Chen, S. Hilmer, B. Shneiderman, and E. P. Hoffman. Interactively optimizing signal-to-noise ratios in expression profiling: project-specific algorithm selection and detection p-value weighting in Affymetrix microarrays. *Bioinformatics*, 20(16):2534–2544, 2004.
- [18] I. M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.
- [19] S. Tarkkanen, K. Miettinen, and J. Hakanen. Interactive poster: Interactive multiobjective optimization — a new application area for visual analytics. In *Proceedings of the 2009 IEEE Symposium on Visual Analytics Science and Technology (VAST 2009)*, pages 237–238, Oct. 2009.
- [20] J. Waser, R. Fuchs, H. Ribičič, B. S. G. Blöschl, and M. E. Gröller. World lines. *IEEE TVCG*, 16(6):1458–1467, Nov.–Dec. 2010.