This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

Visual Parameter Space Analysis: A Conceptual Framework

Michael Sedlmair, *Member, IEEE*, Christoph Heinzl, *Member, IEEE*, Stefan Bruckner, *Member, IEEE*, Harald Piringer, and Torsten Möller, *Senior Member, IEEE*

Abstract—Various case studies in different application domains have shown the great potential of visual parameter space analysis to support validating and using simulation models. In order to guide and systematize research endeavors in this area, we provide a conceptual framework for visual parameter space analysis problems. The framework is based on our own experience and a structured analysis of the visualization literature. It contains three major components: (1) a data flow model that helps to abstractly describe visual parameter space analysis problems independent of their application domain; (2) a set of four navigation strategies of how parameter space analysis can be supported by visualization tools; and (3) a characterization of six analysis tasks. Based on our framework, we analyze and classify the current body of literature, and identify three open research gaps in visual parameter space analysis. The framework and its discussion are meant to support visualization designers and researchers in characterizing parameter space analysis problems and to guide their design and evaluation processes.

Index Terms—Parameter space analysis, input-output model, simulation, task characterization, literature analysis.

1 Introduction

Over the last decade, simulation models have become increasingly prevalent in a variety of application areas. In the visualization literature, for instance, case studies have shown how such simulation models were used to better understand weather and climate phenomena [56], the spread of infectious diseases [1], biological cell profiling [43, 57], and complex engineering and design problems [4, 18, 21, 37]. Structurally, all these examples are based on simulation models that define a set of *parameters* as inputs and are able to compute corresponding *outputs* for a particular parameterization.

From an abstract lens, many examples show recurring structures, tasks and goals. A typical goal is, for instance, the *optimization* of the output by identifying reasonable input parameter settings. Assessing the optimality of outputs often involves trading-off multiple contradicting objectives as well as qualitative judgments of complex data like time series [1, 37], segmented image data [69], animations [18], and 3D geometry [21]. Fully automatic optimization is then often too complex, expensive, or simply not clear how to achieve, and must be complemented by a human manually inspecting simulation outputs.

Traditional approaches of solving such problems were based on *informed trial and error* strategies. Based on prior knowledge and experience, the input parameters are set to a specific value. Then, the model is run and outputs are manually inspected. If the outputs are not satisfactory, the next iteration starts and the model is re-run with a different set of parameter values. A major drawback of this approach, however, is that model runs are often very expensive, that is, it takes minutes or even hours for single runs. In such cases, trial and error leads to severe and unwanted interruptions during the workflow.

To overcome these drawbacks, many researchers have recently proposed more structured workflows. To do so, interesting parts of the parameter space are coarsely sampled to generate input parameter sets. Then, the corresponding outputs are computed offline for all of these sample settings, e.g., over-night or over the weekend. Finally, visualization approaches allow for exploring, investigating, and understanding the space of sampled inputs and their resulting outputs. Such approaches have become known as *visual parameter space analysis* techniques and offer an attractive possibility to deal with the complexity of the models while still keeping the human in the loop.

The current body of work in visual parameter space analysis comprises mostly tools and case/design studies from different application areas [1, 4, 14, 16, 37, 56, 57, 76]. In this paper, our goal is to take a step back from the current application-oriented lens on visual parameter space analysis and provide an abstract *conceptual framework*. The framework can be used to describe, discuss, and evaluate visual parameter space analysis solutions across different application domains, as well as to guide researchers in their design and evaluation decisions.

With our framework, we specifically make three primary contributions. First, we propose a *data flow model* (Section 4) that abstractly describes visual parameter space analysis problems and characterizes recurring data manipulation operations: sampling input parameters, deriving objective measures from outputs, and predicting outputs with cheaper surrogate models. Second, we present a classification of *four navigation strategies* (Section 5). We most importantly distinguish between local-to-global and global-to-local navigation strategies. In local-to-global strategies exploration starts from inspecting a specific sampled simulation run and then provides ways to navigate through other runs. In global-to-local strategies the exploration starts with an overview over all runs and then allows users to drill down into specific runs. Our third primary contribution is a characterization of *six typical analysis tasks* (Section 6) in visual parameter space analysis: optimization, partitioning, fitting, outliers, uncertainty, and sensitivity.

Our framework is based on our own experience working in visual parameter space analysis, collaborations with simulation experts, as well as a structured literature review of case/design studies in this area. This work additionally led us to identify three *open research gaps* to guide future work in this area (Section 8). Within the framework, we also offer a unified set of *definitions and terminology* facilitating research communication and progress. We consider these as secondary contributions of our work.

1.1 Definitions

The set of problems we are focusing on appears in the context of computational *input-output models*. We define input-output models broadly as any sort of function that maps a set of *input parameters* to a set of *outputs*. Together we simply refer to them as *variables*. Input-output models can therefore be, for instance, computational simulations, but most other types of algorithms also match these characteristics.

- *Michael Sedlmair is with the University of Vienna. E-mail: michael.sedlmair@univie.ac.at.*
- *Christoph Heinzl is with the University of Applied Sciences Upper Austria. E-mail: Christoph.Heinzl@fh-wels.at.*
- *Stefan Bruckner is with the University of Bergen. E-mail: stefan.bruckner@uib.no.*
- *Harald Piringer is with VRVis. E-mail: hp@vrvis.at.*
- *Torsten Möller is with the University of Vienna. E-mail: torsten.moeller@univie.ac.at.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014; date of publication xx xxx 2014; date of current version xx xxx 2014. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

A typical goal of using such input-output models is to find an input parameter setting that leads to "good" output results. To achieve this goal, it is necessary to *sample* the model by setting the input parameters to specific values and compute the outputs corresponding to these inputs. One specific sample is also referred to as a simulation *run*; all samples/runs together are referred to as *sampled data*. In some application domains, this sampled data is also referred to as an *ensemble* [56]. Given that, we define *parameter space analysis* as follows:

> Parameter space analysis (PSA) is the systematic variation of model input parameters, generating outputs for each combination of parameters, and investigating the relation between parameter settings and corresponding outputs.

In some cases this process might be achieved fully automatically. Our focus is on how interactive visualization facilitates this analysis. We refer to this concept as *visual parameter space analysis (vPSA)*.

1.2 Example: Tuner

A typical example for visual parameter space analysis is the tool Tuner by Torsney-Weir et al. [69]. One of the *input-output models* in their case is a brain segmentation algorithm. As *input parameters*, this model takes a scanned image of the brain as well as a set of numerical control parameters that define how the algorithm operates. The *output* is a segmented brain image where different brain regions are marked, for instance, as background, skull, white matter, or grey matter.

Running the model with different settings of control parameters results in tremendous variations of the quality of the output segmentation. The goal is therefore to identify a parameter setting that leads to "good" segmentations. In this example, finding a "good" segmentation necessitates to subjectively trade off multiple objectives and fully automatic approaches are not suitable. Thus, Torsney-Weir et al. [69] suggested to coarsely *sample* the parameter space over night and then use methods of visual parameter space analysis to explore and analyze the *sampled data*, that is, instantiations of different input parameter settings and their corresponding output segmentations.

We will use Tuner as a running example for introducing our framework. More details will be discussed along the way, such as Figures 2, 3, 4 and 5 that refer to Tuner. We will also introduce other examples of visual parameter space analysis applications to further illustrate our abstract framework.

2 Background

With the growing amount of published visualization research, building up a higher-level, more theoretical understanding of the work in our field becomes increasingly important. Towards that goal, this paper follows in the line of structured analyses of the visualization literature [11, 31, 39] . Bertini et al. [11], for instance, proposed a systematization and overview of quality measures and derived implications for future work. Here, we focus on a similar goal as Bertini et al. but for the area of visual parameter space analysis.

Our framework specifically focuses on problem abstraction, as well as strategies and tasks that occur in visual parameter space analysis. Many researchers have called for a stronger focus on such task and problem characterizations in visualization research [39, 47, 51, 70]. Following these calls, researchers have recently started to more actively focus on pure *problem characterization* papers. Kandel et al. [34], for instance, have studied analysts within the social and organizational context of companies. Kang and Stasko [35] characterize usage patterns and problems by conducting case studies with their text analysis tool Jigsaw. Earlier work from Tory and Staub-French characterizes visualization practices and collaboration patterns of building designers [72]. Our work follows a similar goal, that is, characterizing a specific set of problems. However, we have a different focus: while the above papers focus on specific application domains, we focus on a specific set of abstract problems, visual parameter space analysis, across application domains.

Previous work on *task characterization* has mostly focused on straightforward low-level tasks [2, 5, 36, 65], such as detecting outliers, or high-level goals [3, 41], such as hypothesis generation. Only recently, researchers have started to characterize complex tasks that lie between these two extremes and that better reflect the needs of real users [17, 47, 58, 61]. Our work on tasks has similar goals. However while the previous work is targeted at generic visualization tasks, we focus on a specific set of data analysis challenges appearing around visual parameter space analysis. We argue that characterizing problems and tasks from more specific angles is indispensable for getting a more concrete understanding of users' needs in these areas.

We see our work between the extremes of narrow, domain-specific task characterizations as done in design studies [64], and generic task taxonomies. Notable examples along these lines are Lee et al.'s work on characterizing tasks for graph visualization [40], and Sedlmair et al.'s work on dimensionality reduction tasks [62].

3 Method

The framework is primarily based on our own experience conducting design studies in parameter space analysis and collaborating with simulation experts in different domains [4, 9, 10, 14, 18, 55, 69]. In these design studies, we started to identify and describe data and task characteristics. Here, we build on these domain-specific experiences, propose an abstract, domain-independent framework, and derive novel insights in terms of analysis strategies, tasks, and open research gaps.

To additionally ground our framework, we conducted a structured, in-depth analysis of the relevant research literature. Our assumption is that these research papers can be seen as a proxy for the problems, data and tasks of end users. From the visualization literature, we gathered an initial set of 112 research papers that we deemed potentially interesting. A closer analysis of these papers led us to a set of *21 core-relevant papers* [1, 4, 9, 10, 14, 16, 18, 21, 26, 37, 43, 44, 45, 46, 55, 56, 57, 68, 69, 73, 76]. This selection was based on a set of exclusion criteria that we defined. First, we specifically excluded papers *without concrete applications* of parameter space analysis. Without a close connection to a concrete application we cannot reliably argue about user tasks. Second, we excluded papers with *automatic analyses only* to keep the focus relevant to the visualization community. Third, we excluded papers that *did not match our definition* of parameter space analysis, as outlined in Section 1.1.

Similar to a machine learning approach, we split the 21 core papers into two groups, a "training" and a "validation" set. We selected and iteratively analyzed 14 papers (training set), with three major rounds of iterations. We used this first round of analysis to step-by-step improve and refine the initial conceptual framework that we developed based on our own experience and collaborations. This analysis specifically informed our characterization of exploration strategies (Section 5), analysis tasks (Section 6), and research gaps (Section 8). We then analyzed the remaining 7 papers to validate the robustness of our framework (validation set). In general, our analysis was inspired by open, axial and selective coding strategies as used in Social Science [19, 24]. Overall, each paper was coded by at least two authors (average 2.8 coders/paper). More details about the methodological approach can be found in the supplemental material.

In the following sections, we will use selected examples from the 21 core-relevant papers to illustrate our framework. Table 1 on page 7 summarizes the final categorization of these 21 papers.

4 Data Flow Model

Our first contribution is a data flow model that depicts how data is generated and manipulated in a visual parameter space analysis setting. Specifically, we characterize three key operations as part of this model: *sampling* the input parameter space, *derivation* of objective measures from the model output, and *prediction* of not-yet-computed (or unsampled) outputs using computationally cheap surrogate models.

4.1 Basic Input-Output Model

The focus of our work is on *input-output models*, such as computational simulation models or algorithms. For a more evocative abstraction of these models, we use a simple graphical representation depicting the data flow.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

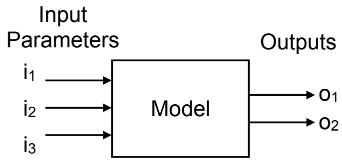


Fig. 1. Simple input-output model with 3 input parameters and 2 outputs.

Consider the simple example in Figure 1. This model takes three inputs and maps them to two outputs. The model might represent a hypothetical weather forecast model that takes current temperature, humidity, and pressure and based on them computes tomorrow's probability of rain, as well as the speed of the wind. Such models can come in very different forms. For instance, *stochastic* models [14, 74] yield different outputs for repeated runs with the same input parameter setting. On the other hand, *deterministic* models [18, 69] produce the same outputs whenever run with the same parameter setting.

In the simplest form, the inputs and outputs of a model come as real numbers. In that case, the model can be represented as a mapping $f : \mathbb{R}^m \to \mathbb{R}^n$, with $m = 3$ and $n = 2$ in the example of Figure 1. However, other data *types* are also common. We broadly classify input and output types into two groups: (1) *multi-variate/multi-dimensional*, and (2) *complex objects* [52]. This distinction is based on a semantic level, not on mathematical concepts.

As in the simple example above, inputs and outputs can come as a *set of semantically meaningful variables* (or dimensions) that are either quantitative, ordered, or categorical. In the literature examples we surveyed, we found that these sets of input/output variables rarely exceeded 100 variables. Moreover, in all cases these variables were semantically meaningful. That is, they were well chosen by the scientists or users that study a specific model. We propose to refer to these input parameters as *multi-dimensional* and to the output characteristics as *multi-variate*. This choice leans on common mathematical terminology [20] and allows us to distinguish between inputs and outputs. Further, we will not refer to these variables as high-dimensional as this is a term usually common in machine learning and statistics where the number of dimensions is in the thousands or millions, and where dimensions have no strong semantic meaning, such as pixel values in an image. Note, that we are not proposing a clear-cut number of variables/dimensions between multi- and high-dimensional, but argue that the strong or weak semantic meaning of dimensions/variables distinguishes these two.

Alternatively, inputs/outputs can come as (semantically) *complex objects*. For example, a 2D/3D image is a single complex object (despite the fact that they can be modelled mathematically as $N \times N$ pixels, or N^2 dimensions). Images cannot be easily described with a single quantitative/ordered/categorical variable. The semantic unit is the complex object itself. Other examples are animations, performance graphs, social networks, or robot behaviors, just to name a few.

Naturally, both can coexist in a model. Consider the running example on image segmentation from the introduction (Tuner). Here, a brain segmentation algorithm takes a scanned image of the brain as input and returns a segmented image as output where different colors mark the individual brain regions, as illustrated in Figure 2.

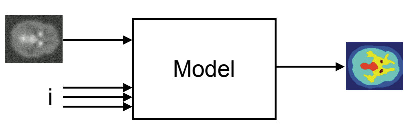


Fig. 2. Model with a complex object in addition to numerical variables as input and a complex object as output. Example from Torsney-Weir et al. [69].

Both the unsegmented 2D input image, as well as the segmented 2D output image are semantically complex objects. Additionally, the model takes some quantitative input parameters that can be adjusted to control the segmentation process. This example highlights the existence of different *classes of input parameters*. To better characterize these differences, we adopt a classification from the statistics community that separates input parameters into three classes [60]:

- *Control parameters* are parameters the user can directly manipulate. These parameters are of primary interest to parameter space analysis problems, such as the three numerical inputs in the example above (Figure 2).
- *Environmental parameters* are parameters that can be measured in the real world, such as the un-segmented brain scan image in the example. They are often prone to small changes and, hence, are modeled as random variables. Therefore, these are parameters that often cannot be directly controlled by the user.
- *Model parameters* are implicit parameters often needed for the numerical realization of the model such as setting certain thresholds, grid spacings or convergence criteria. They might be important during the model building but are mostly hidden during the usage of a model.

4.2 Sampling

At the heart of visual parameter space analysis is the systematic sampling of the input parameter space, and the generation of respective outputs for each sample point (a specific setting of input parameters). Figure 3 shows the sampling process by means of the segmentation example. Each of the parameter settings leads to a different segmentation output.

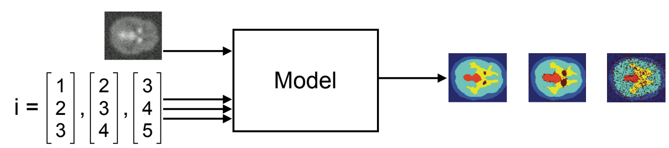


Fig. 3. Sampling a model. Here, 3 different samples are generated by running the model with 3 different input settings.

Most of the papers we analyzed used either regular or stochastic sampling strategies. Regular Cartesian sampling—also known as full-factorial designs in the statistics literature—was the most favored approach. In the case of random sampling, uniform random sampling is used. Some strategies also employ Latin Hypercube approaches. While these sampling strategies allow an overview of behaviors in the parameter space, few tools we analyzed directly supported sampling strategies from within the tools [10, 44, 69, 76]. We refer to this direct integration as *integrated sampling* that allows users themselves to trigger and refine sampling processes, for instance, to generate additional samples or adapt sampling strategies.

4.3 Derivation

It is common that the output of a model is a complex object (18/21 of our analyzed examples). For an effective parameter space analysis, many outputs will have to be studied together requiring an efficient summary, specifically for complex objects. In such cases, the user of the model might want to derive *objective measures* that summarize the essential characteristics of the complex model output. We refer to them as *derived outputs*. Consider the segmentation example again. Figure 4 shows that for each segmented output image, a set of scalar objective measures is computed. In this example, the objective measures are computed by comparing the segmented image to a ground-truth, hand-segmented image. The measures quantify how much the segmented areas differ between the output and the ground-truth image.

Alternatively, the use of pair-wise similarity (or distance) metrics allows an easier comparison of different outputs visually or algorithmically. The distance metric can then be used to provide an overview with distance-based visualization techniques such as MDS plots [29].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

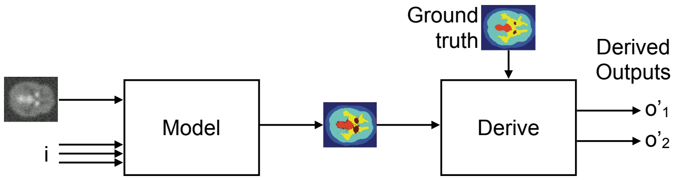


Fig. 4. Derive additional variables. In this case, the segmented output image from the algorithm is compared to a ground truth, hand-segmented image. Differences are quantified as derived outputs.

4.4 Prediction

Sampling a model provides discrete combinations of inputs and outputs. Consider a simple model with 2 input parameters and 1 output. Data from sampling this model 100 times could be easily visualized in a scatterplot with the 2 inputs being the axes, the 100 samples are drawn as points, and the output is mapped to a color-scale which is used to encode each sample point.

However, often the model user is interested in seeing outputs at locations that have not been sampled. If sampling is cheap these points can be just computed on the fly. However, usually generating samples is computationally expensive and would therefore interrupt the analysis process. In these cases, cheap surrogate models [60] can be leveraged to *predict* outputs that have not been sampled by the real model. Moreover, surrogate models might allow one to predict all un-sampled areas and to reproduce the actual continuous-to-continuous mapping between inputs and outputs. Creating continuous spaces from discrete samples is, in the signal processing community, referred to as approximation or interpolation [20]; in the statistical community this is known as regression and prediction [13].

To illustrate this prediction step, let us once again come back to the segmentation example from above. Predicting the two derived objective measures from the three input control parameters leads to a continuous-to-continuous mapping which now can be represented with Hyperslices [75] instead of discrete scatterplots. The three input parameters are mapped to the dimensions, and the two outputs are mapped to orange-white and purple-white color scales, as illustrated in Figure 5. The continuous mapping makes it possible to understand the entire space of relations between in- and outputs, without restricting it to a selected set of sample points.

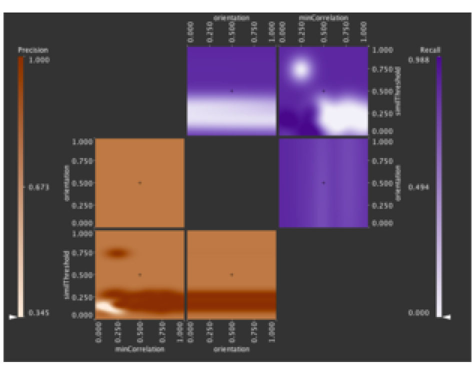


Fig. 5. Hyperslices of the image segmentation algorithm. The 3 input parameters are mapped to the axes. 2 derived outputs are encoded on a white-purple, and a white-orange color scale. Courtesy of Torsney-Weir et al. [69].

4.5 Summary: Data Flow Model

The complete data flow can now be summarized as in Figure 6. The actual input-output model takes multi-dimensional input parameters that can be controlled by the user, and produces *direct outputs* that can either be multi-variate or complex objects. From these direct outputs further *derived outputs* may be extracted. This pipeline can be replaced with a surrogate model taking the same input parameters but now computing *predicted outputs*. Hence, the actual output space contains direct outputs but can include derived and predicted outputs as well. While there is alternative terminology that could describe this problem space, we hope that this data flow model is evocative enough through all areas of visualization research that it will be accepted as a common language.

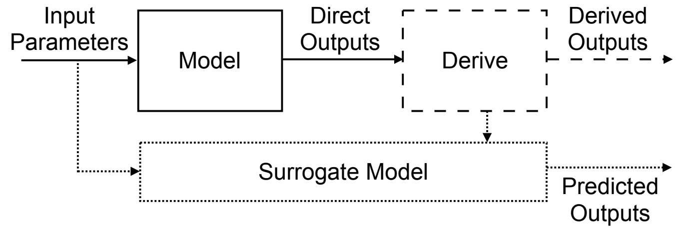


Fig. 6. Our problem abstraction summarized as a data flow model. Dashed and dotted lines indicate the optionality of the additional *derivation* and *prediction* steps.

Note, that this summary depiction reflects a typical scenario. In the real world, more complex scenarios do appear as well, including multiple serial, parallel, or nested derivation steps. However, these can be represented by simply recombining the elementary components of the pipeline. Another interesting question related to this data flow pipeline is where to draw the "line" between the model and derive step. If a model returns multi-variate outputs, oftentimes these have already been "derived" within the model. We argue that this depends on the person who looks at the model and the stage of development. A distinction we find helpful is between visualization researchers and domain experts: direct outputs are what visualization researchers get from domain collaborators, although they might be internally derived; derived outputs are those which visualization researchers actively develop or help developing.

5 Navigation Strategies

When data has been generated via sampling, derivation, and/or prediction, this data needs to be presented to the user for exploration and analysis. Based on our literature analysis, we classify four distinctive strategies of how this data was made available for navigation.

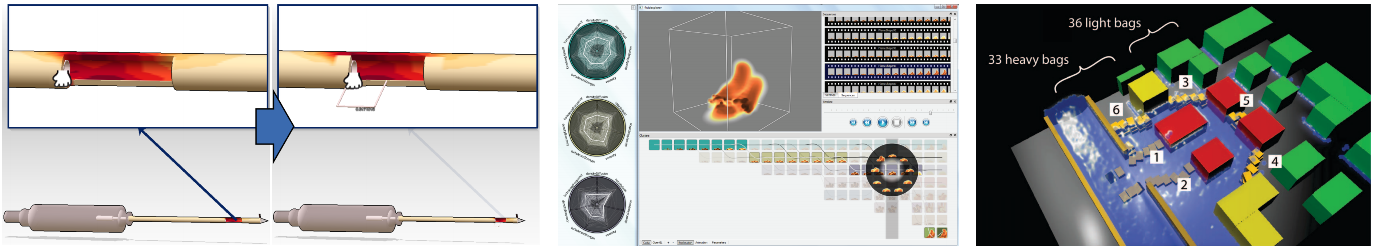
5.1 Informed Trial and Error

Traditionally, parameter space analysis was conducted with *informed trial and error* strategies. Based on prior knowledge, a user (1) runs a model with a specific setting of input parameters creating one sample, (2) inspects the outputs of this sample, and (3) re-runs the model with a refined set of parameter settings if the outcome was not satisfactory. This sequential process can be effective if the simulation output can be produced in real-time. Given that model computations are usually expensive, the informed trial and error strategy, however, has tremendous interruption costs: the user has to wait for minutes, or even hours for new samples to be produced. Usually, this time to find the right parameters is not reported. One simply finds a statement along the lines of "We have found the following parameter settings to yield good results ...". The well known SIFT algorithm [42] serves as a good example. It is a specific feature detector for computer vision applications which works well when a number of parameters are set to specific values. Since no systematic determination has been reported in the paper, it is likely that finding these parameter settings has been done following the wide-spread, traditional trial and error strategy.

5.2 Local-to-Global

To allow for real-time interaction rates despite high model computation costs, researchers have suggested to pre-compute samples before the actual parameter space analysis process. The expensive precomputation can be done, for instance, over night. The column "no. of samples" in Table 1 on page 7 shows the numbers of samples used in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics



(a) local-to-global (b) global-to-local (c) steering

Fig. 7. Examples for different navigation strategies: **(a) Local-to-global**: The user can interactively manipulate the size of the cutting window (input parameters), which is then updating the overlaid stress field heatmap (output). Courtesy of Coffey et al. [21]. **(b) Global-to-local**: The view at the top-right and the view at the bottom show overviews of all simulated explosion (outputs) using representative thumbnail images. Upon selecting one specific explosion its animation can be inspected in the top-mid view. The circular parallel-coordinate plots on the left show the respective input parameter settings. Courtesy of Bruckner and Möller [18]. **(c) Steering**: The user can interactively place sand sacks (input parameters) while a flooding simulation is running (output). Courtesy of Waser et al. [76].

the papers we analyzed. Based on the tradeoff between computational costs on the one hand and analysis accuracy on the other hand, the number of generated samples ranges often between 100 and 1000.

Given this set of precomputed sample points, we identified different characteristic strategies of how they were visually represented and navigated in analysis tools. The local-to-global strategy starts with showing one specific output and lets the user explore alternatives from there. Consider, for instance, a visual parameter space analysis example supporting the design of a medical biopsy device, as shown in Figure 7(a) [21]. Here, a virtual CAD device is used to explore various characteristics such as the length of the tissue cutting window or the outer radius of the cannula, which are the inputs to a simulation model. The simulation output, a scalar stress field, is directly mapped as a heatmap onto the CAD virtual device. The navigation through the pre-computed design space starts with showing a very specific sample, that is, specific device characteristics and a specific stress field. A user can now interactively change the device characteristics (inputs), and in doing so updates the stress field heatmap (output). Step-by-step the user can interactively infer global structures from local searches.

5.3 Global-to-Local

Global-to-local navigation strategies are similarly based on the pre-computation of a large set of sample points. However, instead of starting with a specific sample and navigate alternatives from there, the goal is to start with an overview over all pre-computed samples and then drill-down into more details. In that sense, this strategy is closer to Shneiderman's venerable mantra "Overview first, zoom and filter, then details on demand" [65].

Consider, for instance, how Bruckner and Möller used visual parameter space analysis to support visual effect designers in finding desired explosion animations [18]. Sampling the animation algorithms with different parameter settings, they present interactive thumbnails of clustered animations as shown in Figure 7(b). In doing so, they first reveal the breadth of possible animations to the user, and then support drilling down, identifying and refining good animation candidates.

5.4 Steering

In some cases, a user might want to change the input parameter settings while a simulation runs. We refer to this strategy as *steering*. While the above strategies focus on changing and analyzing control parameters in a systematic way, steering often addresses environmental and model parameters.

We refer to steering environmental parameters as *simulation steering*, which for instance can be found in real-time simulators such as flight or driving simulators. World Lines by Waser et al. [76] is a prime example for this category. As shown in Figure 7(c), their system lets the user place different barriers to contain flooding of a city while the water is rising. Different possible performances can be compared. Users can evaluate alternative scenarios for the assessment of potential hazards by actively steering the simulation while it runs.

On the other hand, steering model parameters refers to on-the-fly adjustment of numerical or other aspects of the computational realization of the model. Examples include changing the grid size or time-stepping parameters. Adjusting these parameters is known as *computational steering* [50].

It is worthwhile to notice that, while we differentiate between local-to-global and steering, others have used the word steering to express local-to-global search [45]. We argue that these two strategies are fundamentally different as one is based on pre-computation or re-sampling (local-to-global), while the other is inherently tied to adjusting parameters during simulation runtime (steering).

6 Analysis Tasks

So far, we have characterized how *data* can be produced from their underlying models, and how *visualization* can support different ways of navigating this data. A third important component is understanding the *tasks* that users eventually want to engage in when doing visual parameter space analysis.

In general, tasks regarding input-output models are often coarsely classified as model building, model validation, and model usage [22]. Our work on visual parameter space analysis primarily focuses on *model validation and usage* tasks for which a computational model already needs to exist. In model validation, modellers question the behavior of the model itself and try to derive formative insights on how to make it better, or summatively judge its performance. In model usage, analysts/scientists use a more or less trusted model without primarily questioning its validity. Visual parameter space analysis of the model can then be used for various purposes. It might be used to guide *design and engineering* processes, for instance, of a biopsy device as discussed above [21]. In *policy making*, decisions can be informed by simulating different "possible futures" [14]. Similarly, a model might be used to study *scientific* phenomena such as bird moving patterns that would otherwise be hard or impossible to study [10]. Or, it simply might be used for *training* purposes emulating real systems in a simulation [76]. While there is no clear-cut line between model validation and usage, we found it a helpful distinction when discussing visual parameter space analysis problems.

With the goal of providing better guidance for visualization researchers and designers, we intended to characterize visual parameter space analysis tasks on a more fine-granular level. Based on our own work and the literature review, we describe a set of six recurring analysis tasks: *optimization, partitioning, fitting, outliers, uncertainty, and sensitivity*. These tasks essentially cross-cut both model validation and usage. Note that it is very common that several of these analysis tasks co-occur in real application scenarios.

6.1 Optimization: *"Find the best parameter combination given some objectives."*

One of the most common tasks is to find an input parameter setting that leads to satisfying output results. Oftentimes objective functions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

can be formulated and numerical measures be derived from the direct outputs respectively. If the objective can be summarized in a single scalar there is a multitude of numerical optimization strategies that can be employed [15].

However, when there are *multiple competing objectives* finding the best output often relies on *subjective human judgement*, a promise that visual parameter space analysis holds. Consider the example of a segmentation algorithm design as supported by our running example Tuner [69]. On the one hand, 12 derived objective measures need to be balanced. On the other hand, it is necessary to subjectively analyze the performance of the segmentation as the objective measures are not fully capturing the expert knowledge.

In some cases optimization might even be a completely subjective process. Consider the above mentioned example of Fluid Explorer in Figure 7(b): in this example, the optimization heavily relies on qualitative assessment of the outputs, the animated explosions. The users are primarily interested in exploring the output space in order to identify a realization that most closely represents their envisioned goals. A similar example is Marks et al.'s venerable work on Design Galleries [44]. Both approaches rely on presenting thumbnails of images or animations which are organized according to a similarity measure.

6.2 Partitioning: *"How many different types of model behaviors are possible?"*

The goal of a partitioning task is to find a partitioning—or clustering, or segmentation—of the output space and relate that back to input parameter settings. In doing so, it is possible to understand what different types of outputs can be expressed by the existing model. A good example is Bergner et al.'s work clustering different fuel cell performance graphs (model outputs), followed by mapping their cluster IDs back into the input space [10]. The input space is shown as a 2D-dimensional scatterplot with the sample points colored according to clusters in the output space. This representation reveals "shapes" of input parameter settings that lead to similar output results.

6.3 Fitting: *"Where in the input parameter space would actual measured data occur?"*

During building and validating a model, it is of interest to see how and whether real measured data can be expressed by the model. In that sense, fitting represents an inverse problem: given model outputs only, what input parameters would yield this behavior? This is also akin to regression analysis in statistics or approximation and interpolation methods in signal processing. While mathematically, this could be formulated as an optimization problem, the user might need a different mind-set and therefore a different visual encoding and interaction design. Improving the understanding of differences between model and reality helps to fit the model more closely to the underlying real world system that is simulated. HyperMoVal [55] is an example that specifically focuses on the validation of regression models. HyperMoVal seeks to support the fitting task by simultaneously plotting the regression model together with known validation data. This approach allows users to analyze how well model outputs align with the real system.

6.4 Outliers: *"What outputs are special?"*

The abstract task of finding outliers can have different specific meanings in model usage and validation. When using a more or less trusted model, it can refer to detecting anomalies in simulations, for instance, to understand interesting and unique phenomena in weather forecast models [56]. On the other hand, when building and validating a model, it can refer to identifying implausible outputs that would not have been possible in an underlying real system. The aforementioned example of HyperMoVal [55], for instance, reports on a case study where an outlier turned out to be an implausible validation sample.

6.5 Uncertainty: *"How reliable is the output?"*

Understanding uncertainties in model usage and validation can come in different forms [22]. In our literature analysis, we specifically identified:

- Aleatoric/statistical uncertainty (lack of knowledge modelled through random variables, often found in environmental variables): "How much do (non-deterministic) runs with the same parameter settings differ?"
- Structural uncertainty: "How much does the model differ from reality?" (a form of epistemic uncertainty)
- Prediction uncertainty (of surrogate models): "How accurate are predicted outputs?" (a form of epistemic uncertainty)

Understanding and integrating uncertainty into scientific, engineering and design processes has gained considerable attention [22]. Yet, the visualization and communication of uncertainty is done cautiously in many systems. Consider, for instance, decision making tools such as Vismon [14], a visual tool for fisheries data analysis. In the Vismon project, the managers and stakeholders (the users of the system) were already overwhelmed with the complexity of the data they need to consider. Hence, the system was developed to bring aspects of uncertainty to the forefront only when explicitly requested by the user. This trend could change as the literacy about sources and quantification of uncertainty sweeps through the different application areas.

6.6 Sensitivity: *"What ranges/variations of outputs to expect with changes of input?"*

Mathematically, sensitivity might be expressed as an uncertainty of the input parameter value, and is therefore often considered a subfield of uncertainty quantification [22]. However, while some of the mathematical approaches of quantifying uncertainty might be applicable to sensitivity analysis, the semantic understanding and articulation of sensitivity is different. Hence, we find it helpful to articulate it as a separate analysis task. In the tools we have studied, sensitivity was never merged or considered a form of uncertainty. In analyzing sensitivity one distinguishes between *global* and *local* sensitivity [59], however, we have only found support for local sensitivity in the tools we have surveyed. Specifically, we have found sensitivity to be cross-cutting through most other analysis tasks:

- Optimization: The question arising is the stability of the output for slight changes of the optimal input parameters. Users are willing to choose a less optimal solution, if it is guaranteed that the solution is stable to small changes of input parameters (specifically, environmental parameters that cannot be directly controlled by the user)
- Partitioning: The question arising here is one of stability of partitions, i.e., how quickly or slowly does one partition change to another when changing the inputs?
- Fitting: Given some specific measured data, the question is how large a range of inputs will yield the model output representing the data measured.

For analyzing sensitivity, it might be useful to *predict* outputs with surrogate models. Reproducing a partial or full continuous-to-continuous mapping between inputs and outputs supports a better understanding of local neighborhoods surrounding points of interests, which in turn is crucial for sensitivity analysis. This approach is, for instance, used in our running example Tuner, in which sensitivity analysis was identified as an important task. Figure 5 shows how HyperSlices were used to navigate the continuous-to-continuous mapping between inputs and predicted outputs.

7 Discussion

Table 1 shows the final result of our iterative analysis of the 21 papers. The cells mark how we classified these papers according to our framework. Naturally, the papers we analyzed were not written with our theoretical lens in mind, necessitating interpretation and in-depth discussions in their analysis. We see our main contribution in summarizing, abstracting, and classifying different characteristics of visual parameter space analysis into a holistic conceptual framework based on these 21 papers.

After reviewing our framework's relation to other theoretical visualization models, we provide guidance on how to use the framework, and discuss its focus and limitations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

		Data Flow Model (Sec. 4)			Navigation Strategies (Sec. 5)			Analysis Tasks (Sec. 6)						Additional details on Model/Data: Input				Direct Outs		Sampling Strategy			No. of Samples				
	No. of Applications presented	Integr. Sampling	Derivation	Prediction	Local-to-Global	Global-to-Local	Steering	Optimization	Partitioning	Fitting	Outliers	Uncertainty	Sensitivity	Multi-D: 1-10D	Multi-D: 11-25	Multi-D: 26-100	Complex Object	Multi-Variate: 1-10	Complex Object	Regular	Stochastic	Other	Hundreds	Thousands	More than 10k	User-defined	
Afzal et al. [1]	2	*	■				■	*									■		■	■						■	Afzal et al.: Epidemic modeling and response evaluation [1]
Amirkhanov et al. [4]	1	*	■			■		■			*		■	■					■	■						■	Amirkhanov et al.: 3DCT ScanPositions [4]
Berger et al. [9]	1	■		■		■		■				P	■		■			■			■			■			Berger et al.: Uncertainty-Aware Exploration [9]
Bergner et al. [10]	3	*	■		*	■		*	■	*	*		*			■			■	■			■				Bergner et al.: Partitioning for Computer Simulations [10]
Booshehrian et al. [14]	1		■	■		■		■				A	*	■					■		■		■				Booshehrian et al: Vismon — Fisheries Management [14]
Brecheisen et al. [16]	3	*	■		■	*		*					■	■					■	■			*not reported*				Brecheisen et al.: Parameter Sensitivity for DTI Fiber Tracking [16]
Bruckner & Möller [18]	1	*	■			■		*	■					■					■		■		■				Bruckner & Möller: Visual Effects Design [18]
Coffey et al. [21]	2			■	■			■		*					■				■		■		■	■			Coffey et al.: Design by Dragging [21]
Guo et al. [26]	1		■		*	■		*		■				■					■	■			■				Guo et al.: Multivariate Linear Trend Discovery [26]
Konyha et al. [37]	2					■		■		*	■		*	■					■	■				■			Konyha et al.: Interactive VA of Families of Function Graphs [37]
Luboschik et al. [43]	1	*	■			■		*		■	■	A	■	■					■			■			■		Luboschik et al.: Simulation trajectories [43]
Marks et al. [44]	3	*	■			■		■							■				■		■		■				Marks et al.: Design galleries [44]
Matkovic et al. [45]	2		■			■		■	*		■		*	■					■	■			■				Matkovic et al.: Common Rail Injection System [45]
Matkovic et al. [46]	1	■			*	■		*			■		■		■				■	■			■	■			Matkovic et al.: Families of Data Surfaces [46]
Piringer et al. [55]	1			■	*	■		*		■	■	S	■	■				■		■			■				Piringer et al.: HyperMoVal [55]
Potter et al. [56]	1		■			■					■	A	*				■		■			■	■				Potter et al.: Ensemble-Vis [56]
Pretorius et al. [57]	2	■				■		*	*	■					■				■	■				■			Pretorius et al.: Image Analysis [57]
Spence et al. [68]	1		■			■		■	■		*			■				■			■		■				Spence et al.: Visualization for functional design [68]
Torsney-Weir et al. [69]	2	■	■	■		■		*		■	*	P	*	■					■	■			■				Torsney-Weir et al.: Tuner [69]
Unger et al. [73]	1		■		■	*			*	■	*	*	*		■				■	■			■				Unger et al.: Validation of Geoscientific Simulation Models [73]
Waser et al. [76]	1	*	■				■	■					*				■		■	■						■	Waser et al.: World Lines [76]

Table 1. The table summarizes the 21 application/design study papers we analyzed in terms of our framework. A cell is marked with yellow when a certain aspect of our *data flow model* is supported by the application. Blue marks indicate the main *navigation strategy* a certain tool follows. Green marks the *analysis tasks* that are primarily supported. Secondary data operations/strategies/tasks are labeled with an asterisk (*). We label a cell as secondary if a strategy or task was not explicitly targeted by the authors but might still be feasible with their tool. In the uncertainty column we further differentiate between aleatoric (A), structural (S), and prediction uncertainty (P), as described in Section 6. Grey shows additional information relevant to visual parameter space analysis.

7.1 Relation to Other Visualization Models

Our framework can be best contextualized using Munzner's Nested Model [51]. The Nested Model organizes the visualization design and validation process into four levels: (1) domain problem characterization, (2) problem/data/task abstractions, (3) visual encoding/interaction design, and (4) algorithm design. Our work in general, and the data flow model (Section 4) and analysis tasks (Section 6) in particular, focus mainly on the abstraction layer (level 2). We ground these abstractions in a thorough analysis (literature analysis and first-hand experience) of domain problems (level 1). We also connect the framework upwards to visual encoding/interaction design (level 3) by characterizing navigation strategies (Section 5).

We also sought to organize our tasks and strategies using the multi-level task typology proposed by Brehmer and Munzner [17]. This typology is organized as why, how and what and presents a set of abstract tasks living in these categories. While we found the general categories of why and how helpful in guiding our analysis, we could not directly match our framework into this typology. Our work addresses analysis tasks specific to visual parameter space analysis that have not been discussed in their typology. We see this fact as a confirmation on the many calls for more work on problem and task analyses [17, 47, 51]. Understanding the richness and variety of visualization problems, and putting them together into a theoretical underpinning remains a major challenge of our community.

7.2 Framework Usage

Our framework can be used in three different ways: (1) descriptive—for describing a significant range of visual parameter space analysis problems and solutions; (2) evaluative—to help assess design alternatives; and (3) generative—to support creating new ideas [7, 8].

Descriptive Usage The terminology we proposed, the data flow model, as well as the analysis task characterizations can be used to abstractly describe domain problems for which visual parameter space analysis solutions are generated. We refined and validated our framework by describing visual parameter space analysis applications from 21 papers, and are therefore confident that the framework will be descriptive for many other application examples as well.

We anticipate three major benefits when describing visual parameter space analysis work through our abstract lens. First, it will help in problem-driven research, such as design studies [64], to abstractly characterize the problem and translate domain knowledge into actionable design decisions. Second, technique-driven researchers can use it to clearly characterize their goals and assumptions. Third, the framework then can facilitate the communication between researchers. Specifically, it will allow for an easier mapping between problem-driven and technique-driven work. Additionally, it will allow to compare and relate findings *across different application domains*, accelerating progress in visual parameter space analysis research in general.

Evaluative Usage The navigation strategies and analysis tasks we characterized will help to better assess multiple alternatives in designing visual parameter space analysis tools.

Consider the example of deciding between *local-to-global* and *global-to-local* navigation strategies. Global-to-local starts with a broad overview over many/all possible model outcomes, while local-to-global starts from a specific output and then allows the interactive exploration of alternatives. From that perspective global-to-local seems more powerful in many cases. However, this decision might interact with other factors. For instance, deep immersion into specific decisions might outweigh global exploration of alternatives in certain situations. Also, such decisions depend on how complex the model output is and how easy/hard it is to computationally *derive* objectives and/or visually provide an evocative overview.

As intrinsically true for all conceptual frameworks, these identified strategies are naturally a simplification of the reality. For real tools we found that aspects of local-to-global and global-to-local navigation were often combined with different views supporting different strategies. We marked these combinations with an asterisk in Table 1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

However, having a clear characterization of strategies and tasks helps to better reason about choices and eventually make more informed design decisions.

Generative Usage Finally, our framework can also be used to generate and inspire new ideas. We believe that the framework is concrete enough to depict visual parameter space analysis problems and solutions, yet general enough to inspire other areas as well.

While we use our input-output data flow model to reason about simulations and algorithms it could, for instance, be similarly used to describe and analyze the visualization process itself. The idea would be to generate, that is, *sample* many different visual encodings and then use *derived* quality measures to spot interesting ones. Some of the pioneering work in visualization includes the study of transfer functions for volume rendering (Design Galleries [44], visualization spreadsheets [32], parallel-coordinate-style interfaces [71], VisTrails [6]), the analysis of the rendering pipeline [38], graphs [33], and the analysis of multi-variate data projections [66, 77]. Our framework gives a new theoretical lens to think about this line of work and might be used to generate new ideas that have not been thought of with the traditional perspective.

7.3 Focus and Limitations

Our work is grounded in our own experience working in visual parameter space analysis, as well as a structured analysis of 21 core-relevant papers. This approach comes with standard limitations of qualitative theory building [19]. Selecting and coding papers, as well as generating the framework was inevitably shaped by our previous experience.

To keep the effort of our in-depth literature analysis manageable, we selected 21 core-relevant papers with a specific focus based on our definition of visual parameter space analysis. Nevertheless, there are many other papers that are closely related to our endeavor, which is reflected in the larger set of 112 papers that we initially gathered (see supplemental material for a full list).

For instance, we specifically focused on visualizing relations of inputs and outputs sampled from computational models. However, also *measured data* often comes in a similar form of two groups of related variables. In statistics, they are usually called independent (analogous to our inputs), and dependent variables (outputs). Consider, for instance, Guo et al.'s work on sensitivity analysis, which is a task that also appears in our framework. Their focus is on previously measured data. The example in their paper relies on a benchmark dataset of measured diamond weight, color, clarity, and cut (independent var.), and their relation to price (dependent var.) [27].

We further selected core-relevant papers with a focus on the investigation of input-output relations. Other visualizations of simulation data focus mainly on representing the *output space*. Nocke et al., for instance, primarily look at solutions of how to visualize complex outputs from climate simulations [54]. Given the complexity of representing even individual climate simulation outputs, they only marginally focus on input-output relations. Smith et al. address the question of morphing between shape objects resulting from computational design models, such as car CAD models [67]. Their work is closely related to our *prediction* strategy.

We mainly focused on model validation and usage tasks of an existing computational model. We did not explicitly include other *model building* tasks, such as feature extraction, selection, or transformation. Consider, for instance, Mühlbacher's and Piringer's work that discusses how visualization can support building regression models [49].

Finally, we specifically set out to study *visual* parameter space analysis. While this focus was intentional, we believe that our framework might be useful for more general parameter space analysis scenarios with a less substantial role of visual encodings as well. We also believe that the framework is general enough to be useful for the closely-related areas discussed above, although these lines of work were not part of our core literature analysis. Naturally, all possible generalizations cannot be tested in a single paper. Validating, refining, and extending our framework to include other problem areas is an interesting step for future work.

8 Research Gaps and Future Work

Through our practical and theoretical work on visual parameter space analysis, we additionally identified three research gaps that are described in the following. We believe that these gaps provide ample opportunity for future work.

8.1 Data Acquisition Gap

As described above, the visual parameter space analysis pipeline starts with *sampling* the parameter space. All following analysis steps rise and fall with this crucial first step. However, only a few current tools directly support sampling from within the tool (4/21 papers address it as a primary goal). Reasons for not supporting sampling might include potential engineering and organizational hurdles when integrating the model with the visualization tool [63], high computational costs of the sampling process, and the fact that proper sampling of multi-dimensional spaces is not trivial. Given that model computations are usually expensive, sampling multi-dimensional spaces and categorical parameters pose challenges. Often, it is not clear which sampling strategy to utilize and simple uniform random sampling becomes the default, without a deeper understanding of its implications. For instance, not every instance of a random distribution truly assures a uniform covering of the space [53].

While such decisions might not pose any challenges to domain scientists with a strong mathematical background, they do for others without this background. Ingram et al. classified the latter as middle-ground users [28]. These users would tremendously benefit from easy-to-use visual parameter space analysis tools that integrate the sampling step and help reveal underlying sampling assumptions and implications.

8.2 Data Analysis Gap

Integrating computational analysis methods into the visualization pipeline also poses a major challenge for future work. While the *predict* step in our pipeline refers to the challenge of building good surrogate models, the *derive* step deals with how to derive good objective measures. While domain knowledge is crucial to be successful, there are a number of general strategies that have been developed in the data analysis communities of statistics and machine learning. Yet, they are similarly important for visual analysis.

18/21 of our analyzed papers described models with complex objects as direct outputs. For those, deriving is a crucial step to open up more sophisticated visual analysis approaches. Deriving fosters more holistic and powerful global-to-local analysis strategies. Deriving also better supports most of the tasks we characterized in our work, such as multi-objective optimization, partitioning, or sensitivity analysis.

The actual model in our pipeline is usually a black box to us as visualization researchers (and also might be to the domain experts themselves). In contrast, we argue that we need to better understand methods for deriving and predicting. Making these steps a white box to us will allow us to better support a much richer set of analyses steps, and help to make them accessible to middle-ground users as in the DimStiller project [28].

The visualization community is already very active in this area, for instance, by focusing on quality measures for multi-variate data representations [11, 66, 77]. Also, many examples we analyzed already utilize derived measures (15/21 papers). Given the richness of potential model outputs, however, we deem this only as a starting point for an important area of future work.

8.3 Cognition Gap

Another major challenge is how to facilitate the cognitive understanding of and navigation through multi-dimensional spaces. As humans we are inherently 3D plus time beings. Naturally, understanding higher dimensions seems inherently impossible. While this challenge is shared with general multi-dimensional visualization, visual parameter space analysis comes with specific characteristics that are important to understand.

Consider, for instance, multi-objective optimization. Pareto front visualizations have been found to be helpful for such endeavors [69].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

A Pareto front basically connects all solutions where no objective measure can be improved without degrading another one, and therefore gives useful constraints for output space navigation. Yet, while visual Pareto fronts are straight-forward in 2D [69], it is not clear how to visually depict or even efficiently compute them in higher dimensions. Vismon [14], for instance, samples specific multi-dimensional options for direct comparison, not taking advantage of a higher-dimensional Pareto front.

It is also not clear how many objectives in an optimization problem can be cognitively handled by humans. Is this number, for instance, following the general 7 ± 2 rule of capacity limits in human information processing [48]? A possible approach might be Gleicher's work on generating projections according to the users' needs [25]. Here the user builds an analysis system one dimension at a time, allowing the gradual increase of complexity.

8.4 Other Areas of Future Work

Beyond these gaps, conducting research in visual parameter space analysis gives ample opportunity to study many previously identified visualization challenges. *Scalability* considerations are inherently part of many computational model analyses necessitating out-of-core, parallel, and cluster computing solutions [12]. On the other hand, the rich set of potential visual and computational analysis methods for parameter space analysis problems calls for good concepts of user *guidance* [28]. Sophisticated *provenance* approaches [23] could help in this regard to better track what parts of the parameter space have already been explored and which not. Especially policy making examples such as Vismon [14], inherently involve multiple stakeholders, giving ample opportunity to study *collaboration* processes [30]. Eventually, we also need a stronger focus on *user evaluation* [39]. Analyzing the current literature revealed that most parameter space analysis applications were evaluated with usage scenarios. These scenarios make clear how data could be analyzed, but leave out how users actually used these tools themselves. Some notable exceptions, such as Pretorius [57], give richer usage descriptions, where actual users used the tool and reported anecdotal evidence [51].

Finally, we want to echo previous calls on the importance of problem-driven work such as case and design studies [47, 64]. Our work is grounded in the first-hand experiences reported in 21 of such application papers. Deriving our higher-level theoretical framework would not have been possible without this problem-driven work.

9 Conclusion

We have presented a conceptual framework that characterizes the data flow, navigation strategies, and analysis tasks in visual parameter space analysis problems. We hope that our work will establish a useful abstraction of otherwise domain-specific concepts and will propel the fascinating area of visual parameter space analysis to a fruitful area of further visualization research.

Acknowledgments

We wish to thank Alireza Ghane, Eduard Gröller, Tamara Munzner, Niki Popper, Thomas Torsney-Weir, Steven Bergner, as well as the anonymous reviewers for their helpful feedback. This work was supported in part by FWF-funded project Nr. P 24597-N23 (VISAR), the K-Project "Non-Destructive Testing and Tomography Plus" (FFG), and the COMET K1 program of the Austrian Funding Agency (FFG).

References

[1] S. Afzal, R. Maciejewski, and D. Ebert. Visual analytics decision support environment for epidemic modeling and response evaluation. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pages 191–200, 2011.

[2] R. A. Amar, J. Eagan, and J. T. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symp. on Information Visualization (InfoVis)*, pages 111–117, 2005.

[3] R. A. Amar and J. T. Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE Trans. on Visualization and Computer Graphics*, 11(4):432–442, 2005.

[4] A. Amirkhanov, C. Heinzl, M. Reiter, and E. Gröller. Visual optimality and stability analysis of 3DCT scanpositions. *IEEE Trans. on Visualization and Computer Graphics*, 16:1477 –1487, 2010.

[5] N. Andrienko and G. Andrienko. *Exploratory analysis of spatial and temporal data*. Springer Berlin, Germany, 2006.

[6] L. Bavoil, S. Callahan, P. Crossno, J. Freire, C. Scheidegger, C. Silva, and H. Vo. VisTrails: Enabling interactive multiple-view visualizations. In *IEEE Conf. on Visualization*, pages 135–142, 2005.

[7] M. Beaudouin-Lafon. Designing interaction, not interfaces. In *Conf. on Advanced Visual Interfaces (AVI)*, pages 15–22. ACM, 2004.

[8] B. B. Bederson and B. Shneiderman. *The craft of information visualization: readings and reflections (Chapter 8)*. Morgan Kaufmann, 2003.

[9] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911 – 920, 2011.

[10] S. Bergner, M. Sedlmair, T. Möller, S. Nabi Abdolyousefi, and A. Saad. ParaGlide: Interactive parameter space partitioning for computer simulations. *IEEE Trans. on Visualization and Computer Graphics*, 19(9):1499–1512, 2013.

[11] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.

[12] J. Beyer, A. Al-Awami, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. ConnectomeExplorer: Query-guided visual analysis of large volumetric neuroscience data. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2868–2877, 2013.

[13] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.

[14] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum*, 31(3):1235–1244, 2012.

[15] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

[16] R. Brecheisen, A. Vilanova, B. Platel, and B. ter Haar Romeny. Parameter sensitivity visualization for DTI fiber tracking. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1441–1448, 2009.

[17] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.

[18] S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1467–1475, 2010.

[19] K. Charmaz. *Constructing grounded theory. A practical guide through qualitative analysis*. Sage Publications, Inc, 2006.

[20] W. Cheney and W. Light. *A Course in Approximation Theory*. American Mathematical Society, 2009.

[21] D. Coffey, C.-L. Lin, A. Erdman, and D. Keefe. Design by dragging: An interface for creative forward and inverse design with simulation ensembles. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2783–2791, 2013.

[22] Committee on Mathematical Foundations of Verification, Validation, and Uncertainty Quantification; Board on Mathematical Sciences and Their Applications, Division on Engineering and Physical Sciences, National Research Council. *Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification*. The National Academies Press, 2012.

[23] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Engineering*, 10(3):11–21, 2008.

[24] D. Furniss, A. Blandford, and P. Curzon. Confessions from a grounded theory PhD: Experiences and lessons learnt. In *ACM Trans. Computer-Human Interaction (ToCHI)*, pages 113–122, 2011.

[25] M. Gleicher. Explainers: Expert explorations with crafted projections. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2042–2051, 2013.

[26] Z. Guo, M. O. Ward, and E. A. Rundensteiner. Model space visualization for multivariate linear trend discovery. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pages 75–82, 2009.

[27] Z. Guo, M. O. Ward, E. A. Rundensteiner, and C. Ruiz. Pointwise local pattern exploration for sensitivity analysis. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pages 131–140, 2011.

[28] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2014.2346321, IEEE Transactions on Visualization and Computer Graphics

Dimstiller: Workflows for dimensional analysis and reduction. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pages 3–10, 2010.

[29] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel MDS on the GPU. *IEEE Trans. on Visualization and Computer Graphics*, 15(2):249–261, 2009.

[30] P. Isenberg, N. Elmqvist, J. Scholtz, D. Cernea, K.-L. Ma, and H. Hagen. Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization*, 10(4):310–326, 2011.

[31] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Möller. A systematic review on the practice of evaluating visualization. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2818–2827, 2013.

[32] T. Jankun-Kelly and K.-L. Ma. Visualization exploration and encapsulation via a spreadsheet-like interface. *IEEE Trans. on Visualization and Computer Graphics*, 7(3):275–287, 2001.

[33] T. Jankun-Kelly, K.-L. Ma, and M. Gertz. A model and framework for visualization exploration. *IEEE Trans. on Visualization and Computer Graphics*, 13(2):357–369, 2007.

[34] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2917–2926, 2012.

[35] Y.-A. Kang and J. T. Stasko. Examining the use of a visual analytics system for sensemaking tasks: Case studies with domain experts. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2869–2878, 2012.

[36] P. R. Keller and M. M. Keller. *Visual cues: practical data visualization (Appendix B)*. IEEE Computer Society Press, 1993.

[37] Z. Konyha, K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual analysis of families of function graphs. *IEEE Trans. on Visualization and Computer Graphics*, 12(6):1373–1385, 2006.

[38] J. Kronander, J. Unger, T. Möller, and A. Ynnerman. Estimation and modeling of actual numerical errors in volume rendering. *Computer Graphics Forum*, 29(3):893–902, 2010.

[39] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Trans. on Visualization and Computer Graphics*, 18(9):1520–1536, 2012.

[40] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proc. BELIV*, pages 1–5. ACM, 2006.

[41] Z. Liu and J. T. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):999–1008, 2010.

[42] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE Conf. on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[43] M. Luboschik, S. Rybacki, F. Haack, and H.-J. Schulz. Supporting the integrated visual analysis of input parameters and simulation trajectories. *Computers & Graphics*, 39(0):37 – 47, 2014.

[44] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design Galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH*, pages 389–400, 1997.

[45] K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering—rapid visual prototyping of a common rail injection system. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1699–1706, 2008.

[46] K. Matkovic, D. Gracanin, B. Klarin, and H. Hauser. Interactive visual analysis of complex scientific data as families of data surfaces. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1351–1358, 2009.

[47] M. Meyer, M. Sedlmair, P. S. Quinan, and T. Munzner. The nested blocks and guidelines model. *Information Visualization*, pages 1–16, 2013.

[48] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[49] T. Mühlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[50] J. D. Mulder, J. J. van Wijk, and R. van Liere. A survey of computational steering environments. *Future generation computer systems*, 15(1):119–129, 1999.

[51] T. Munzner. A nested model for visualization design and validation. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[52] T. Munzner. *Visualization Analysis and Design (Chapter 2)*. A K Peters Visualization Series. Taylor and Francis / CRC Press, 2014. to appear.

[53] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Society for Industrial and Applied Mathematics, 1992.

[54] T. Nocke, M. Flechsig, and U. Böhm. Visual exploration and evaluation of climate-related simulation data. In *Winter Simulation Conf.*, pages 703–711, 2007.

[55] H. Piringer, W. Berger, and J. Krasser. HyperMoVal: Interactive visual validation of regression models for real-time simulation. In *Computer Graphics Forum*, pages 983–992, 2010.

[56] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. Johnson. Ensemble-Vis: A framework for the statistical visualization of ensemble data. In *IEEE Conf. on Data Mining Workshops*, pages 233–240, 2009.

[57] A. Pretorius, M.-A. Bray, A. Carpenter, and R. Ruddle. Visualization of parameter space for image analysis. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2402–2411, 2011.

[58] R. E. Roth. An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2356–2365, 2013.

[59] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, Ltd, 2008.

[60] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, 2003.

[61] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A design space of visualization tasks. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2366–2375, 2013.

[62] M. Sedlmair, M. Brehmer, S. Ingram, and T. Munzner. Dimensionality reduction in the wild: Gaps and guidance. Technical Report TR-2012-03, Dept. of Computer Science, University of British Columbia, 2012.

[63] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization*, 10(3):248–266, 2011.

[64] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[65] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symp. on Visual Languages*, pages 336–343, 1996.

[66] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.

[67] R. Smith, R. Pawlicki, I. Kokai, J. Finger, and T. Vetter. Navigating in a shape space of registered models. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1552–1559, 2007.

[68] B. Spence, L. Tweedie, H. Dawkes, and H. Su. Visualization for functional design. In *IEEE Symp. on Information Visualization (InfoVis)*, pages 4–10, 1995.

[69] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.

[70] M. Tory. User studies in visualization: A reflection on methods. In *Handbook of Human Centric Visualization*, pages 411–426. Springer, 2014.

[71] M. Tory, S. Potts, and T. Möller. A parallel coordinates style interface for exploratory volume visualization. *IEEE Trans. on Visualization and Computer Graphics*, 11(1):71–80, 2005.

[72] M. Tory and S. Staub-French. Qualitative analysis of visualization: A building design field study. In *Proc. BELIV*, pages 7:1–7:8. ACM, 2008.

[73] A. Unger, S. Schulte, V. Klemann, and D. Dransch. A visual analysis concept for the validation of geoscientific simulation models. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2216–2225, 2012.

[74] A. Unger and H. Schumann. Visual support for the understanding of simulation processes. In *IEEE Pacific Visualization Symp.*, pages 57–64, 2009.

[75] J. J. van Wijk and R. van Liere. HyperSlice: Visualization of scalar functions of many variables. In *IEEE Conf. on Visualization*, pages 119–125, 1993.

[76] J. Waser, R. Fuchs, H. Ribicic, B. Schindler, G. Blöschl, and E. Gröller. World lines. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1458–1467, 2010.

[77] L. Wilkinson, A. Anand, and R. L. Grossman. Graph-theoretic scagnostics. In *IEEE Symp. on Information Visualization (InfoVis)*, pages 157–164, 2005.