

Otto-von-Guericke University, Magdeburg

Faculty of Computer Science



Bachelor Thesis

Interactive Visual Analysis of Gear Stress as Surface Ensemble

Author:

Lena Cibulski

July 11, 2016

Advisors:

Prof. Dr.-Ing. Holger Theisel

Department of Simulation and Graphics, Otto-von-Guericke University, Magdeburg

Priv.-Doz. Dr. Krešimir Matković

Interactive Visualization Group, VRVis Research Center, Vienna

Cibulski, Lena:

Interactive Visual Analysis of Gear Stress as Surface Ensemble

Bachelor Thesis, Otto-von-Guericke University, Magdeburg, 2016.

I hereby declare that I have authored this bachelor thesis independently and without the use of publications and resources other than those explicitly stated in the references.

Magdeburg, July 11, 2016

LENA CIBULSKI

Scientific Environment

The presented thesis has been submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computational Visualistics to the Department of Simulation and Graphics at Otto-von-Guericke University, Magdeburg.

I would like to thank *Holger Theisel* for supervising this thesis and for providing constant support in all concerns throughout the development process.

The thesis has been developed as part of an internship at the VRVis Research Center in Vienna under the guidance of *Krešimir Matković*.

In this context, I want to express my thanks for his valuable input and feedback that contributed to this thesis as well as for introducing me to the world of publications. He was always available for encouraging discussions and has also generated data and sample images for the illustrative lighting example.

The research presented has been produced in collaboration with *Borislav Klarin* and *Martin Sopouch*, both from AVL List Ltd. in Graz, Austria. As experts in the domain of gearing systems, they provided data and analysis tasks for the presented case study and also participated in its evaluation.



FAKULTÄT FÜR
INFORMATIK



zentrum für
virtual reality und visualisierung
forschungs-gmbh



Abstract

Computational simulation is widely used for investigation of dynamic real world processes in various fields of application. Scientists and engineers often intend to find appropriate input parameters that produce a desired output. Studying the sensitivity of simulation models for this purpose involves the execution of multiple simulation runs, which result in an ensemble of simulation output and corresponding input parameter settings. In addition to scalar values, ensemble data can also contain complex output, such as data surfaces, which arise in the context of meteorological data, as temperature or humidity measurements over an area, or related to powertrains, as forces on gears and bearings, for example. Accordingly, there is a need for interactive visual analysis approaches to directly include such complex data types in the analysis process to cope with the increasing size and complexity of ensemble data sets. Traditional approaches mostly consider each dimension to be a scalar value, limiting analysis of multiple-run simulation data to characteristics that can sufficiently be formalized by scalar dimensions. The interactive visual analysis approach presented in this thesis supports a thorough investigation of *surface ensembles*, i.e. collections of two-dimensional functions across several simulation runs that are considered as atomic data unit.

To address the challenging task of simultaneously analyzing a large number of data surfaces, we introduce novel aggregation techniques and corresponding visualizations and advance existing methods towards a more informative representation of many surfaces. Three-dimensional multi-resolution box plots, hierarchical heat maps, intersection contours, and a surface display allow for aggregation and visualization of surface ensembles at different levels. Together with standard views, these techniques are integrated into a coordinated multiple views framework that supports feature selection via brushing, focus+context visualization, and on-the-fly data derivation. We evaluate the proposed methodology in terms of a case study from the field of gear transmission design, where domain experts simulate contact stresses that arise on the tooth flanks during operation. In this domain, the presented techniques add significant value to traditional gear contact analysis approaches by enabling simultaneous exploration of a large number of different gear designs. Positive feedback and reported analysis speed-up indicate the usefulness of the presented methodology.

Contents

1	Introduction	1
2	Related Work	6
3	Interactive Visual Analysis of Surface Ensembles	11
3.1	Data Model	11
3.2	Illustrative Example	15
3.3	Interactive Visual Analysis of Surface Ensembles	18
3.3.1	Three-dimensional Box Plots	21
3.3.2	Two-dimensional Box Plots	26
3.3.3	Heat Maps	29
3.3.4	Intersection Contours	32
3.3.5	Three-dimensional Surface Display	35
3.4	Implementation and System Integration	37
4	Case Study: Advanced Gear Stress Analysis	39
4.1	Components and Working Principles of Gears	40
4.2	Simulation and Data Conversion	43
4.3	Interactive Visual Analysis of Gear Stress	50
5	Conclusion	59
6	Future Work	61
A	Appendix	63
	Bibliography	66

1. Introduction

Simulation is a reliable tool to understand the complexity and characteristics of real world phenomena [Fis95]. It allows for investigation of underlying mechanisms that control the behavior of physical phenomena in nature, science, and technology by using models to describe them. A model represents observable quantities of a real world system. It takes a set of variables, called input parameters, which are processed by a piece of software, the simulation solver. Simulation in turn produces output, which characterizes the behavior of the modeled system. Such output can be complex, such as time series or surfaces [BSM⁺13]. The simulation process is completed by an analysis of the results, which goes along with visualization [SB11].

Understanding how simulated processes evolve depending on varying input parameters delivers insights into the complex inter-dependencies between input and output variables. Determining regions in input parameter space that lead to qualitatively different model behavior is important for a deeper understanding of the simulated system.

Additionally, we are interested in not only determining such parameter settings, but also studying why some of them lead to desired output, while others produce undesired results. More precisely, two main issues that analysts, engineers, and scientists are interested in can be summarized as: (1) How does the output change as the values of input parameters are varied? Having a simulation model, this is not difficult to answer. (2) What is the optimal input parameter configuration so that simulation produces a desired output? In this case, an optimal solution is not necessarily the setting that produces the globally best output. Instead, we search for a parameter configuration that is acceptable and at the same time stable and nonsensitive to slight deviations of one or more of the input variables. This means that parameter configurations that are generated through slight variation of the original setting should produce output of similar quality. We want to avoid cases in which a small variation of an input parameter suddenly produces output that is not acceptable at all.

The issue of finding such practicable parameter settings, however, is not at all trivial to answer. It refers to the reverse of the simulation process, which allows for the determination of input parameters needed to achieve a selected output. Mathematically modeling the inversion of simulation holds various limitations. Numerical rounding errors, sensitivity to the choice of parameters, numerical instability as well as accuracy issues present difficulties [MS00].

Instead, *ensemble simulation* can be used to study the sensitivity of simulation models and to gain a deeper understanding of the investigated system. Multiple simulation runs are executed with varying input parameters for a given case scenario. Simulation output is computed for each combination of input parameters. This results in a collection of simulation runs consisting of specific values for input parameters and corresponding output. Remember that simulation solvers generate output ranging from simple scalars, to curves and other complex data types, such as surfaces or volume data.

Using interactive visual analysis, we can establish and investigate hypotheses about the interplay of input parameter configurations and simulation results. It also facilitates the exploration of information that emerge from advanced simulation as well as the discovery of patterns and tendencies in complex data sets. In this way, interactive visual analysis methods help to gain deeper knowledge about the simulated system and related physical phenomena.

Problem Statement

As technologies and computing power have rapidly improved over the last years, more and more tasks that used to be fulfilled by humans are now solved by automatic systems. There are multiple application areas where they are faster, more accurate, or more reliable than human workers. Automatic data analysis is commonly based on statistics, machine learning, or data mining, for example. Such approaches usually take numerical values as input. Therefore, complex output of ensemble simulation is often broken down into numerical features for further processing by means of algorithms. Automatic analysis methods are well-suited for tasks like filtering according to concrete criteria, but as system and task complexity grow such processing alone is not sufficient.

It is therefore reasonable to combine it with interactive visual analysis in order to benefit from the strengths of both interactive and automatic data processing. This results in a semi-automated analysis process, where most effective results are achieved by combining the respective characteristic abilities of humans and computers [KAF⁺08]. However, even state-of-the-art approaches in the field of interactive visual analysis of simulation ensembles propose the use of aggregates together with techniques, such as histogram, scatter plot, and parallel coordinates, that process only scalar values.

For example, LiteVis [SOL⁺15], a simulation system that supports decisions in lighting design, uses a ranking view as well as standard views, such as parallel coordinates, bar charts, and spreadsheets, for analysis of simulation results. All of these techniques use scalar values as input. The original simulation output, the distribution of illuminance values, is not part of the analysis process. Instead, it is only considered for checking purposes by means of a detailed view and for comparison of multiple simulation runs.

Thus, in the context of ensemble data, traditional approaches in visual analysis as well as automatic systems consider each dimension to be a scalar value, such as numerical, categorical, and nominal ones, and usually do not support complex simulation output in its original state.

However, sometimes characteristics of data cannot be sufficiently formalized to be successfully processed by approaches mentioned above. Human perception and recognition of complex data structures cannot be replaced by formalization. The ability to quickly and efficiently register complex information is essential for gaining an understanding of complex systems, which approaches considering only scalar values do not achieve. An analyst can easily see if a surface is uncommon, but it is difficult to formally describe what makes it unusual. Thus, based on numerical values, it is very hard to interpret complex data such as surfaces. For efficient exploration of large ensemble simulation data sets it is required to directly include complex data in the analysis. Adequate visualization techniques that present ensemble simulation data in a visual form that allows analysts to interact with it reduce the cognitive demands of analysis tasks [Kei01]. Multiple-run simulation is employed for understanding and design of dynamic systems across many domains. Accordingly, there is a need for appropriate interactive visual analysis techniques that provide direct access to visualizations of the original complex data and therefore allow to include them in the analysis process.

Goal and Contributions of the Thesis

We aim to provide an interactive visual analysis approach that supports exploration and investigation of multidimensional multivariate data including surface ensembles. The presented approach has not the intention to replace traditional automatic approaches. Instead, it is seen as an efficient additional method, which can help gaining insights from a different point of view. A combination of both directions has the potential to solve highly complex tasks [MGJH15]. We intend to use interactive visualization to improve the investigation of surface ensembles emerging from advanced ensemble simulation, which is characterized by varying parameter settings and observation of corresponding output. Moreover, we aim to enable a deeper understanding of parameter settings that produce desired or undesired output and why they do so. This goes along with choosing an optimal input parameter combination for a given design task.

Therefore, we provide an appropriate methodology for the analysis of multiple simulation runs that includes human recognition of complex data structures in the analysis process. This is realized by presenting complex data, such as multi-dimensional ensemble simulation output, in their original state, for example as surfaces, allowing analysts to use their perception in order to gain insights into the data. Such procedure offers the potential of revealing correlations that would otherwise not be identified.

As we focus on interactive visual analysis of ensemble simulation data, individual simulation runs and their comparison are essential for analysis tasks. To address this issue, we choose a simulation-run centric approach for data management. Based on a multiple-run simulation, we obtain a data set by collecting the set of input parameter

configurations and the corresponding output in a data table. Each row of the table is related to an individual simulation run consisting of input parameters and corresponding output data. In addition, it can also contain derived data, such as distance metrics or other features, that were extracted from a simulation output. Each data record also corresponds to exactly one visual primitive in employed visualizations. In that way, we preserve a consistent data model throughout the entire analysis procedure, from data storage to visualizations to interaction handling. It also addresses the analyst's intuitive mental model of how data is organized, supporting his train of thought and therefore reducing the cognitive load while analyzing complex data sets.

In order to realize the simulation run centric approach, we introduce complex data as atomic attributes focusing on surfaces. A *data surface* is a two-dimensional function f , which is stored in one data table cell. The function domain is represented by a two-dimensional grid, where we have a function value $f(x, y)$ for each of the grid points. If we consider a lighting design example, one data row representing one simulation run, contains numerical values, such as the illuminating power of luminaries, as well as data surfaces, like the distribution of light on a surface, for example. In this case, the surface consists of sample points across the measurement surface, making up the function domain, and illuminance values as function values. All data surfaces in a set of simulation runs representing one output attribute form a *surface ensemble*.

Visualizing and analyzing one such data surface is relatively straightforward. In contrast, when it comes to a large number of data surfaces it is challenging to gain spatial understanding and to identify similarities and differences among the shape and characteristics of the surfaces simultaneously [AWH⁺12].

The proposed interactive visual analysis approach addresses the ambitious task of visualizing and exploring data surface ensembles. For this purpose, in addition to state-of-the-art techniques, we integrate various levels of aggregation: (1) Statistical aggregation is carried out in three-dimensional space using multi-resolution box plots. They show statistical measures for a certain section of the surface domain, but also contain a histogram inside the quartile range. To support detailed inspection and to provide context information, we add a view displaying two-dimensional box plots for further subdivisions of a box plot's section. (2) Hierarchical heat maps at three different levels of detail allow for a side-by-side comparison of two sets of surfaces, but also support detailed investigation of a single ensemble member. (3) Intersection contours of data surfaces with a horizontal cutting plane provide deeper insight into the shape of surfaces and allow for investigation of surfaces from another perspective. (4) A surface view, which displays surfaces as such, enables detailed investigation of a small number of surfaces. Display properties concerning lighting and rendering mode (wireframe or solid shaded) as well as line-width, color, and transparency of brushed and context surfaces can be adjusted to the analyst's needs.

All views are linked and brushes are updated immediately.

Structure of the Thesis

Based on the motivation and leading aim as mentioned previously, the thesis is organized as follows:

Chapter 2 presents relevant work related to visualization of large high-dimensional data sets, interactive visual analysis with respect to complex data types as well as ensemble simulation and visualization.

Chapter 3 deals with the interactive visual analysis approach that is provided to deal with surface ensembles. An easily understandable example serves as basis for introducing the methodology proposed in this thesis. The chapter covers data handling as well as developed techniques. To support the depicted research tasks, aggregation of data surfaces at different levels is provided. Multi-resolution box plots, hierarchical heat maps, intersection contours, and surfaces as such in combination with standard views can be employed simultaneously to enable a thorough analysis of ensemble simulation data. Afterwards, the integration of the presented approach into a coordinated multiple views system is described.

Chapter 4 provides the evaluation by means of a case study addressing the analysis of gear stress simulation. The structure and working principles of gears as well as the simulation parameter and results are introduced. Feedback of domain experts is provided and achievements as well as limitations concerning requirements of gear stress analysis are described.

Chapter 5 and Chapter 6 conclude the thesis results and outline remaining challenges to be addressed in the future.

2. Related Work

The topic of this thesis mainly intersects with the research area of ensemble visualization, but also relates to interactive visual analysis of large high-dimensional data sets, interactive parameter space exploration, and coordinated multiple views. This chapter gives an overview of scientific work performed in the context of the named research fields as well as other work related to this thesis.

Our intention is to gain a deeper understanding of the interrelations between simulation input and output. This goes along with the purpose of finding input parameter settings that produce a desired result. Traditionally, such problems are solved by single-run simulation and trial and error strategies. As this is time consuming and interferes with the workflow, newly proposed methods compute sets of input parameter combinations and corresponding output and use visualization techniques to investigate and explore them. Such methodology is called *visual parameter space analysis* [SHB⁺14].

The underlying sets of input and output is called ensemble and can be computed using multiple-run simulations, which result in output for each specified combination of input parameters. Therefore, visual analysis as suggested by visual parameter space analysis approaches needs to include *ensemble visualization*.

When dealing with such large and complex data sets, it is essential to combine the storage capacity and computational power of computers with the perceptual abilities of humans [Kei01]. Techniques that realize this issue are called *interactive visual analysis*. An efficient data analysis and exploration using interactive visual analysis also calls for appropriate *interaction techniques* that allow analysts to gain a deeper understanding of the presented information.

Visualization is a proven methodology for investigation of huge amounts of multi-dimensional multivariate data, which arise in sciences and everyday life. Exploring such data sets relies on data being presented in a visual form that supports analysts in drawing conclusions [Kei01]. The body of literature dealing with visual exploration and analysis of high-dimensional data is accordingly large.

When interactively exploring multidimensional multivariate data, coordinated multiple views support the perception of complex relations by simultaneous investigation of various visual representations using linking and brushing. In his survey on coordinated multiple views, Roberts summarizes the state-of-the-art of related topics, such as data processing, view generation, and coordination [Rob07]. He also presents opportunities and challenges that come along with coordinated multiple views.

The system WEAVE provides an environment for interactive visualization applications [GRW⁺00]. Prior multiple coordinated views systems focus on existing information visualization techniques and allow the user to choose for representations and linkage [NS00, RLS⁺96]. Unlike them, WEAVE provides integration of custom three-dimensional components into an environment of statistical representations, such as histograms, scatter plots, and parallel coordinates. All views are linked so that coloring in one view is reflected in all other representations.

Matković et al. present ComVis, a tool that supports rapid prototyping and testing of new visual analytics techniques [MFGH08]. It is more flexible, as it simplifies adding new views and supporting new data types. Brushing is also taken to a higher level, as it additionally supports composite brushes, e.g. brushes that are combined using Boolean operations. The handling of analysis sessions is simplified, because the current analysis status can be saved and loaded again later. ComVis serves as environment for the implementation of the methodology proposed in this thesis.

Another important aspect of interactive visualization is the preservation of context information when data subsets of interest are shown in detail. Focus and context (F+C) visualization primarily appeared in information visualization and has made use of different enlargements factors for displaying different parts of the data. This is realized using various distortion techniques, like bifocal display [SA82] or graphical fish-eye [SB92]. Hauser extends the commonly used F+C approaches to scientific visualization [Hau06]. He establishes a generalized definition and uses various graphics resources, such as space, opacity, color, and frequency, to distinguish between focus and context.

A highly interactive focus and context approach that enables the presentation of thousands of function graphs at the same time is presented by Muigg et al. [MKO⁺08]. They establish three hierarchical focus levels as well as one context level that result from individual selections and logical combinations of selections. Image-based algorithms, such as shading and line integral convolution, are combined with a color representation of levels for better recognition and differentiation.

Interaction techniques, such as the described coordinated multiple views as well as focus and context approaches, are extensively used for interactive visual analysis dealing with parameter space exploration and ensemble data resulting from advanced simulations. As these areas of research are closely related to the topic of this thesis, we present the current state-of-the-art in visual parameter space exploration and ensemble visualization below.

A recent survey on visual parameter space exploration is provided by Sedlmair et al. [SHB⁺14], who establish an abstract data model for problem description, describe strategies for data representation and exploration, and present analysis tasks related to

parameter space analysis. They also provide a literature review and identify research gaps. In the following paragraphs, we refer to the most relevant approaches, relating to visual parameter space analysis as well as interactive visual analysis of ensemble data, that guided the development of the methodology presented in this thesis.

ParaGlide is a visualization system presented by Bergner et al. that allows for interactive exploration of parameter spaces of multidimensional computer simulations [BSM⁺13]. Their research corresponds to our goals as they intend to divide the input parameter space of a simulation into partitions of parameter settings that represent similar output behavior. This contributes to investigation of model complexity, sensitivity, and qualitative comparison. ParaGlide also supports analysts in understanding the complex interplay of simulation input and output by providing insights into the impact and significance of parameters. However, the system is limited to building a pre-stage for parameter optimization, as it only provides a contextual basis for choosing parameter configurations. The relatively generalized approach and abstract data model address various application domains, but at the same time lacks views that are specific to data types and applications.

As ensemble simulation is increasingly used for investigation of various phenomena, there is a growing need for appropriate visual analysis approaches that are able to cope with the resulting data. An overview of characteristics of ensemble data and consequences for visual analysis is provided by Wilson and Potter [WP09]. They also identify general leading questions, which apply to various fields of application.

Matković et al. propose an interactive visual analysis approach for exploration of correlations in injection simulation data [MJJ⁺05]. For investigation of simulation results, they extract scalar response parameters from time-dependent fuel injection curves, which describe the opening and closing of the injector. To search for optimal control parameter settings, they use scatter plots, histograms, and parallel coordinates, combined with a linking and brushing approach.

More complex data is handled by LiteVis, which was mentioned in Chapter 1 [SOL⁺15]. As a reminder, Sorger et al. present a system that supports visualization and comparison of illuminations for different case scenarios based on simulation. Simulation output comprises an illumination value for each texel in the given scene. The analysis of simulation results, e.g. ranking of simulated lighting configurations, is performed based on scalar indicators, such as investment costs, maximum illuminance, and others. Local and global indicators are weighted, which allows users to turn their attention to certain properties. Interactively changing the weights allows for exploration of different preferences. This simulation ranking was inspired by LineUp, a visualization technique that uses bar charts for visualization of multiple-attribute rankings [GLG⁺13].

However, none of these approaches includes the original complex data structure in the data analysis. Instead, each dimension is considered to be a scalar value, so that common multivariate visualization techniques can be applied. This is not appropriate for a powerful exploration of large high-dimensional data sets, because it does not take advantage of the human perceptual ability of efficiently registering complex information.

In contrast to traditional approaches, there are various methods that consider complex data types, such as curves and surfaces, as complex internal data structure. They present techniques that directly integrate complex data structures into the analysis process and support interactive visual analysis specific to ensembles of such data.

1D functions, commonly in the context of time-series data, arise in many different fields of study. Konyha et al. present an approach to the interactive visual analysis of function graph ensembles [KMG⁺06]. They extend the approach coping with injection simulation data [MJJ⁺05] by a more complex internal data structure and advanced interactive methodology. Function graphs are included in data records as an individual unit. The proposed analysis procedures are attuned to this specific type of data. Standard views as well as a curve view, which shows a family of function graphs simultaneously, allow for detection of patterns, tendencies, and outliers in the data. However, visual clutter can impair the perception when the number of function graphs is too large to be handled using approaches like density mapping. Linking and brushing is realized using coordinated multiple views and different brushing methods. This approach is limited to ensembles of 1D function data. However, there are various analysis tasks dealing with complex data sets including surface ensembles.

An intuitive approach of coping with ensemble data is to reduce complexity by summarizing distribution characteristics via statistical measures. Kao et al. present a case study, which deals with visualization of distributions from ensemble data that is based on satellite images [KDP01]. They model uncertainty in ensembles of maps, which represent geophysical variables, such as percent forest cover, for example. For visualization and analysis of such ensembles, they provide scalar pixel-wise summaries per grid cell as well as feature-wise summaries that produce clusters of similarly-valued grid cells. Potter et al. present Ensemble-Vis, which deals with aggregation of weather ensembles [PWB⁺09]. Spatial information across ensemble members at one point in time is presented in the form of statistics mapped to color, height, or contours. Several visual representations are combined to display multiple statistical values. In contrast to that, we propose a visual representation that allows us to show multiple statistical measures across ensemble members at once.

In this thesis, we focus on directly integrating data surfaces in the visual analysis. Common approaches often perform drill down based on scalar aggregates, to which complex data are cut down. In contrast to them, we allow for drill down that is carried out directly on the surface data. That allows to take human perceptual abilities into account and to directly integrate the user's cognition into the analysis process.

Piringer et al. present an overview display, which arranges members of 2D function ensembles as rectangular icons in a scatterplot-like way [PPBT12]. Color encodes function values across the two-dimensional domain. Brushing results in larger icons representing selected functions, which are integrated into the overview of ensemble members. Occlusion is addressed by appropriate data-driven force-directed placement and automatic icon scaling. We advance this approach by providing a hierarchical arrangement of heat maps that allows for better scalability. Nevertheless, the scatterplot-like overview offers promising features that could be used on a medium level of the analysis process, after

the user has already reduced the data to a smaller number of records.

Matković et al. facilitate in-depth analysis of surface ensembles by providing a multiple-level approach, which integrates scalar aggregates (0D) and 1D profiles for each ensemble member as well as surfaces as such (2D) [MGKH09]. Overview levels providing function aggregations can be used for identification of overall trends, before the analyst drills down to closely examine data subsets of interest in more detail. All views are linked and the user can constantly switch between levels during an analysis session. Data surfaces are introduced as an atomic unit in data records. Aggregation is realized per ensemble member, not across members. Standard profiles like minimum and maximum projection are introduced as an extension of scalar aggregation. This allows for investigation of surface behavior with respect to a selected independent variable which simple aggregation does not enable.

This approach closely relates to our work. We additionally provide novel aggregation techniques like multi-resolution box plots, heat maps, and intersection contours, which enhance the visual exploration and analysis of surface ensembles.

As simulation is widely used to investigate physical phenomena in nature and science, the data sets used for the presented interactive visual analysis approaches originate from such a background in large part. This does not mean that interactive visual analysis of ensembles is limited to such kind of data. It is not only applicable for physical data, but also suitable for various other application domains.

Padua et al. present an approach for interactive exploration of parameter space, applied to the issue of comprehending the predictive quality of decision trees in data mining [PSMD14]. They include scalar values, curves, and decision trees as atomic units in data records and provide views that are specific to these complex data types. This integrates the user’s perception into the analysis process and allows her to steer the exploration in the right direction. It also enhances the depiction of cross-parameter relations and simplifies the comparison and analysis of decision trees at different levels of detail. Their work is closely related to our intentions, but the data model is not applicable to our goal of analyzing 2D function ensembles.

In our work we provide a multi-level approach with linking and brushing, based on the well-known methodology of coordinated multiple views, which is integrated into an existing tool, ComVis [MFGH08]. Novel aggregation techniques across ensemble members allow for an accurate investigation of surface ensembles at a certain level of abstraction. In contrast to various common approaches that mostly use scalar dimensions, we directly integrate complex data structures, such as curves and surfaces, into data analysis. Traditionally used color and transparency for differentiation of focus and context data are part of our methodology, but we also introduce a further implementation of the focus and context model to be used with the newly proposed aggregation techniques.

3. Interactive Visual Analysis of Surface Ensembles

Interactive visual analysis of surface ensembles allows for exploration and investigation of large high-dimensional data sets that emerge from advanced multiple-run simulation. It enables the analyst to deal with the increasing complexity of data and to gain insights into such data where traditional approaches are limited. However, the visual analysis of surface ensembles poses various challenges that come along with the large number of dimensions and data records as well as the complex composition of data.

In this chapter, we describe the interactive visual analysis approach proposed in this thesis. Section 3.1 addresses the data model that forms the basis for the proposed methodology, which in turn is in detail described in Section 3.3. In Section 3.2 an illustrative example, which is used to introduce the novel aggregation techniques, is presented. Finally, we demonstrate how the approach has been integrated into a co-ordinated multiple views system in Section 3.4.

3.1 Data Model

As outlined in Chapter 1 and Chapter 2, we intend to directly integrate complex data structures into data analysis. One essential step of visualization is the mapping of a data table to visual representations. It is important that these visual structures represent all data in the data table without additionally showing unwanted data, such as relationships that are not contained originally in the data [CMS99]. This requires an appropriate data model that works well with the desired visualization and exploration techniques. In our case, the data model is supposed to simplify the organization and handling of data and, in particular, to support visual analysis of complex data types. In the context of their research concerning visual analysis of data surfaces, Matković et al. propose a data model that satisfies the named criteria and serves as a basis for the approach presented in this thesis [MGKH09]. It will be described in detail subsequently.

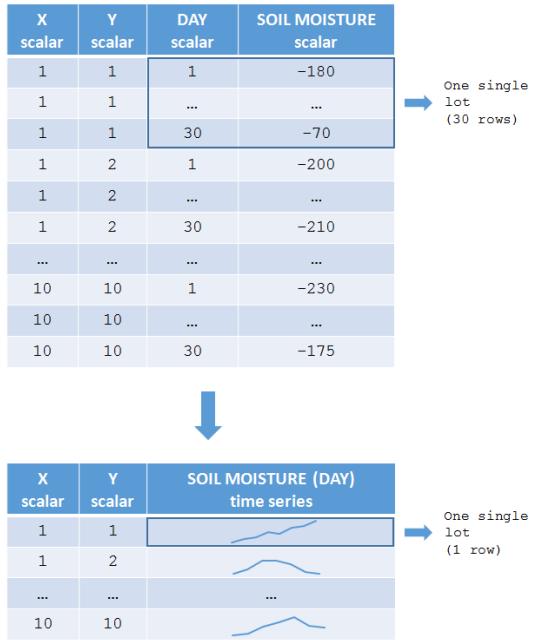


Figure 3.1: Aggregation of data points with scalar dimensions to lot-centric data points containing time series.

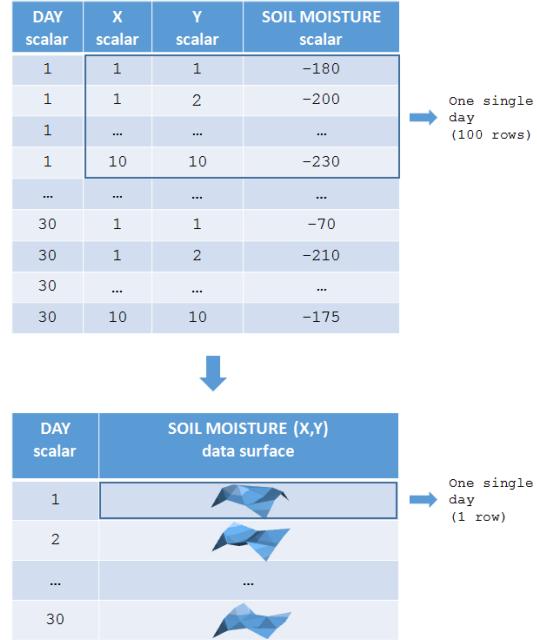


Figure 3.2: Data points for a fixed day are aggregated to a data surface $f(x,y)$.

Data sets are a collection of data points, where each data point is a collection of attributes. As mentioned before, common approaches consider these attributes to be scalar values, for example categorical or numerical.

As an illustrative example, let us consider a farmer owning a field that is divided into 10x10 lots. For the time period of one month (30 days), he measures the soil moisture for each of these lots each day. Using only scalar dimensions, we would represent the data corresponding to one lot as 30 data points with the independent variable *day* and the dependent output dimension *soil moisture* (Figure 3.1, top frame). For the complete field, which consists of 100 lots, this would result in a table consisting of $10 * 10 * 30 = 3000$ rows. It contains three independent control parameters, *x* and *y* determining the position of a single lot and *day*, which differentiates the rows for one specific lot, as well as *soil moisture* as output dimension (Figure 3.1, top).

Such a large data table can be analyzed using coordinated multiple views together with parallel coordinates and a scatter plot, for example, which are suitable for investigating correlations between different parameters. However, such an approach is not appropriate for analysis of how output attributes evolve over time, because the coherency of time steps belonging to the same set of independent variables is not given.

Using data transformations, for example in the form of aggregation or classing, we can encode patterns in the dimensions of a data table and use visualization to make encoded as well as further patterns perceptible [CMS99]. Therefore, we transform the collection of data points described above by deriving structure and values.

As we can see in Figure 3.1, for every 30 data points two of the independent variables

stay the same, as these rows all belong to a single lot with fixed position. We merge all original data points that have the same combination of x and y and thus belong to one lot and obtain one row for one lot, containing a time series dimension, which replaces the merged data points. The time series dimension has *day* as independent variable and *soil moisture* as dependent variable and contains all pairs of values that were originally stored using 30 rows. As depicted in Figure 3.1, bottom, the resulting data set now contains 100 data points instead of 3000 – one data point for each lot. This is just one way to aggregate various of the original data points to a more complex data type. Depending on the leading questions of the application, substructures other than time series may be required. For the example data set, we can also build up substructures of two independent variables as two-dimensional functions $f(x, y)$, also called data surfaces.

For that purpose, we divide the data set into groups of data points that have the same value for the attribute *day*. Within each group, we have 100 data points with different combinations of the control parameters x and y . We can now establish the output dimension *soil moisture* as a function of two independent variables, x and y . In this way, the variable *day* becomes the independent variable. The resulting data table now consists of 30 rows – one for each day – containing the attribute *soil moisture* as a function $f(x, y)$ that represents the spatial distribution of soil moisture for a fixed day.

We can identify different ways for integration of complex data types as atomic units, because data points can be aggregated based on various (selected) dimensions. Depending on the attributes that were chosen for data aggregation, we are able to observe data sets from a certain perspective. Each of the data tables, obtained by data transformations, reveals a different aspect of the stored information and may provide different insights. This supports specific analysis tasks and application questions, as the data model can be defined accordingly.

If we consider the farmer's example again, there are different objectives the farmer wants to accomplish by analyzing the soil moisture: (1) What is the most promising lot judged by the soil moisture observed within a month? For answering this question we use a lot-centric approach, which means that we have one data point for each lot (Figure 3.1). This is useful, because we have to compare multiple lots and examine the soil moisture over time for each of them. (2) On which days should I have additionally watered my field, because the moisture content of the ground was not optimal? This can be solved using a day-centric data model, which holds the distribution of moisture content across the field for each of the 30 days (Figure 3.2). The farmer can compare different days and search for surfaces with soil moisture values that are too low.

Up to now, we considered only one field that is observed over time, resulting in one single surface attribute. Let us assume that the farmer owns several fields, on which he measures soil moisture over time. Then, the day-centric data model can be extended by additional surface attributes – one attribute for each observed field.

Generally, we can describe data organization by a two-level data model, with data points of a certain number of (complex) dimensions at the top level and substructures representing time series or data surfaces by a collection of pairs or triples at the bot-

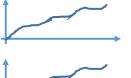
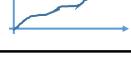
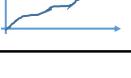
ID	Scalar 1	...	Scalar N	Time Series 1	...	Time Series N	Surface 1	...	Surface N
1	23.3	...	193		...			...	
2	49.6	...	87.2		...			...	
...
M	32	...	121.5		...			...	

Table 3.1: Structure of the general data model. Atomic attributes include scalars, time series, and surfaces. Surface ensembles are obtained by an attribute-centric (blue) or run-centric (orange) approach.

tom level. This allows for reorganization of the data set, resulting in a relatively small number of data points providing the entire information content. An overview of the data model’s general structure is depicted in Table 3.1.

Such a data model also opens up new opportunities for the visualization and exploration of multivariate, multidimensional data sets. Analysts are able to cope with the increasing complexity of data and can gain insights into the complex interrelations between data dimensions, which are not possible otherwise [PPBT12].

As introduced in Chapter 1, we usually build up surface ensembles by considering one specific dependent variable across multiple simulation runs, which we call *attribute-centric* approach (Table 3.1, blue). However, if our data set contains multiple attributes that are of a surface data type, there is another way to look at the data: we can also obtain a surface ensemble by considering multiple surface attributes for a fixed simulation run. We call this a *run-centric* approach (Table 3.1, orange). Let us take a look back at the farmer, who observes soil moisture on several fields over 30 days. Evaluating soil moisture for a certain field over multiple days equals an attribute-centric approach, which results in a surface ensemble for this specific field (across several days). In contrast to that, the run-centric approach yields one surface ensemble for each day, where each ensemble contains one data surface for each of the observed fields.

The representation for both attribute-centric and run-centric approaches is the same: a set of surfaces. As a consequence, both approaches can be used for analysis with the proposed visualization techniques without limitations. The analyst can also constantly switch between these two modes during analysis, allowing her to gain insights into the data from another perspective. In this way, the focus can be arbitrarily chosen to meet the current analysis tasks, which ensures flexibility and a comfortable user experience.

Depending on the origin of data, they often contain errors or missing values that must be coped with before visualization and analysis can start. Additional information can also be added by computing new values from the original ones, as it is done with statistical calculations, for example. Thus, data tables often contain derived values [CMS99]. As described in [KLM⁺12], traditional analysis systems are limited to filtering for data manipulation. For this reason, more complex procedures, such as data derivations, have

to be performed in advance. As analysts often do not know which data will be needed for analysis, the choice of appropriate attributes for derivation prior to analysis poses a challenging task.

To overcome this issue, we provide on-the-fly data derivation, which allows for generation of data derivatives at any time during the analysis process. Computed values are dynamically added to the underlying data table as a new dimension. It can also be used to aggregate individual surfaces by deriving time series or scalars using simple statistical measures. This considerably compresses the representation of a single surface, but at the same time surface characteristics are nearly completely lost. Therefore, we do not consider such aggregations in the following section.

3.2 Illustrative Example

We describe an illustrative example from the lighting simulation domain, which is used to introduce the methodology proposed in this thesis. The example is kept deliberately simple, which is why it does not have a significant reference to common applications. Although it is not intended to be used as a real case study, it is highly useful as an easily understandable illustrative example. We later describe a more detailed case study, which deals with the analysis of stresses in gearing systems and also involves domain experts for evaluation.

An adequate illumination of the workplace is essential for productivity, the absence of stress, and the general well-being at work. Where natural light is not sufficient, for example during winter or at night, additional light sources, such as ceiling luminaries, desk lamps, or floor lamps, have to be provided for an even lighting with, at best, a neutral color. The necessity of a proper illumination is demonstrated by legislative regulations and standards that determine the lighting conditions to be satisfied at working environments [ASc94, ASt98]. As the luminous intensity depends on the scene that is to be illuminated, there is no general rule of how many luminaries with how much illumination power are required for proper illumination. In this case, multiple-run lighting simulation can help with the exploration of various lighting setups for a specific scenario when aiming to find the optimal composition of luminaries.

Similar to the methodology presented in LiteVis [SOL⁺15], we simulate various luminary setups for a three-dimensional scene, which was specified in advance. The simulation computes an illuminance value measured in lux for each sample point of a defined measurement surface in the scene. The distributions of illuminance on the measurement surface can then be analyzed using interactive visual analysis in order to find a reasonable lighting setup that satisfies the specified conditions.

For realization of the simulation we use Radiance, a lighting simulation tool that allows for calculation of light propagation in a specified 3D scene [LS04]. For this purpose, we first define the input of the simulation, such as scene geometry as well as properties like surface materials, lights, and textures. An image of the illuminated scene composed in this way can be viewed after compilation. Afterwards, we execute the solver, which computes illuminance values across the measurement surface as described above.



Figure 3.3: An overview of the scene set-up that is used for simulation of light distributions.

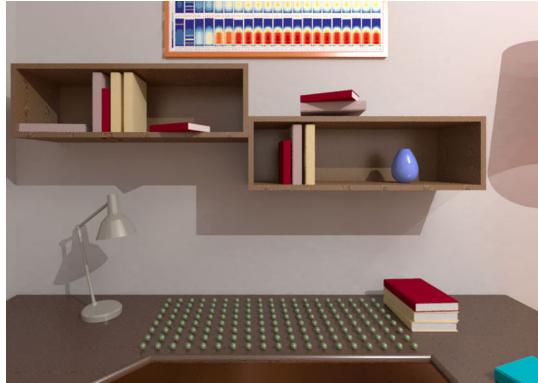


Figure 3.4: The measurement surface consisting of 152 measuring points represented by green spheres. For each of these points illuminance is computed.

Figure 3.3 presents the scene setup that was used for simulation of light distribution. It consists of a room containing a desk, on which a desk lamp is positioned, in front of a window. A floor lamp besides the desk as well as a ceiling lamp provide ambient light. A window in southward direction opens the view onto Vienna, which we chose for localization of the scene. Blinds are mounted in front of the window to allow for control and therefore variation of incident daylight. The measurement surface for simulation is defined as part of the desk surface. It covers an area of 35 by 90 centimeters, which is sampled at a resolution of five centimeters. This results in 152 sample points, as can be seen in Figure 3.4.

We simulate the light distribution across the measurement surface under different lighting conditions. Properties of the lamps and blinds as well as varying daylight serve as control parameters for the simulation. Table 3.2 shows the parameters and the corresponding sets of possible values. We consider all possible combinations of the control parameter values for simulation, yielding $6 * 3 * 2 * 2 * 8 = 576$ different control parameter settings, where one simulation run is executed for each setting. In this way, we obtain 576 light distributions, which result from simulation under different lighting conditions. Luminaries are independent from each other, which means that illuminations for single luminaries can be added up – comparable to a linear combination – to obtain illuminance values for a certain combination of luminaries. We are aware that, in a real application scenario, multiple simulation runs are not required due to this property. Still, as the described example is simple to understand, we use it for explanatory reasons to introduce the proposed methodology.

Figure 3.5 shows rendered close-ups of the illuminated workplace that originate from simulation under eight selected control parameter settings. We can clearly notice the effects of the different control parameter values: generally less light in the morning and evening (b and g), evenly distributed light coming from the ceiling lamp (a and h), shadow stripes originating from halfway closed blinds (c, e, and f), and focused light

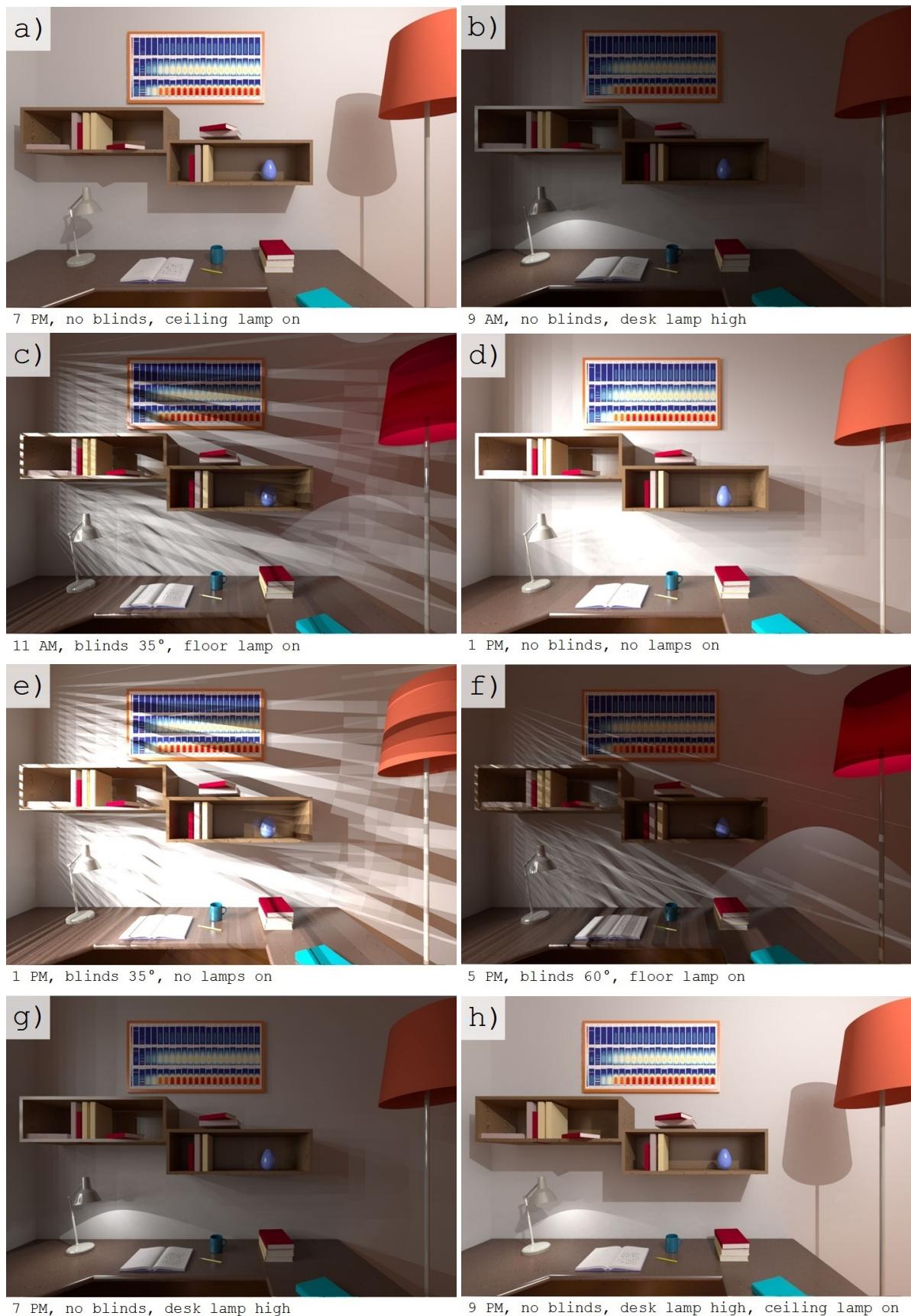


Figure 3.5: Illumination of the workplace under different lighting conditions.

Control parameter	Values
Blinds	none, 0 degrees, 10 degrees, 35 degrees, 60 degrees, 80 degrees
Desk Lamp	off, medium, on
Ceiling Lamp	off, on
Floor Lamp	off, on
Daylight	7 am, 9 am, 11 am, 1 pm, 3 pm, 5 pm, 7 pm, 9 pm

Table 3.2: Control parameters, that were used for simulation of light distributions in an office. All possible combinations of values serve as 576 different lighting conditions.

due to the desk lamp being switched on (b and g). As can be seen from the sample images, some of the control parameter combinations are more suitable for adequate illumination, by providing enough and evenly distributed light, than others, where results are generally too dark, contain shadow stripes, or only parts of the area are sufficiently illuminated.

Using the results of lighting simulation, we obtain one data surface for each simulation run, i.e. a fixed input parameter variation. Each data surface consists of a collection of surface points, which correspond to the sample points and illuminance values on the measurement surface in the scene. In this way, the dimensions of a data surface can be defined as follows: the positions of sample points on the plane measurement surface result in x - and y -coordinates of surface points. The corresponding computed illuminance values are then displayed on the vertical z -axis. When observing multiple runs, i.e. the simulation results of different lighting setups, we obtain a surface ensemble.

3.3 Interactive Visual Analysis of Surface Ensembles

Analyzing the data surfaces resulting from advanced multiple-run simulations is a stressing task for experts from many application domains. While it is relatively straightforward to perform an analysis of one such data surface, the visualization and interactive exploration of a large number of surfaces at once is highly challenging.

However, visual data exploration and analysis of surface ensembles offers great potential for gaining new insights and a deeper understanding of investigated systems. In their work, Piringer et al. identify various analysis tasks that domain experts face when working with advanced ensemble simulation [PPBT12]. This includes sensitivity analysis and optimization of designs on a high level of abstraction as well as comparison of individual simulation results and details-on-demand on a low level of abstraction.

Thus, advanced techniques for visual data exploration and analysis are needed that enable the analyst to complete the named tasks by means of analyzing a large number of data surfaces simultaneously.

The analysis of surface ensembles follows different steps commonly used in interactive visual analysis. First, the analyst is interested in the discovery of overall tendencies

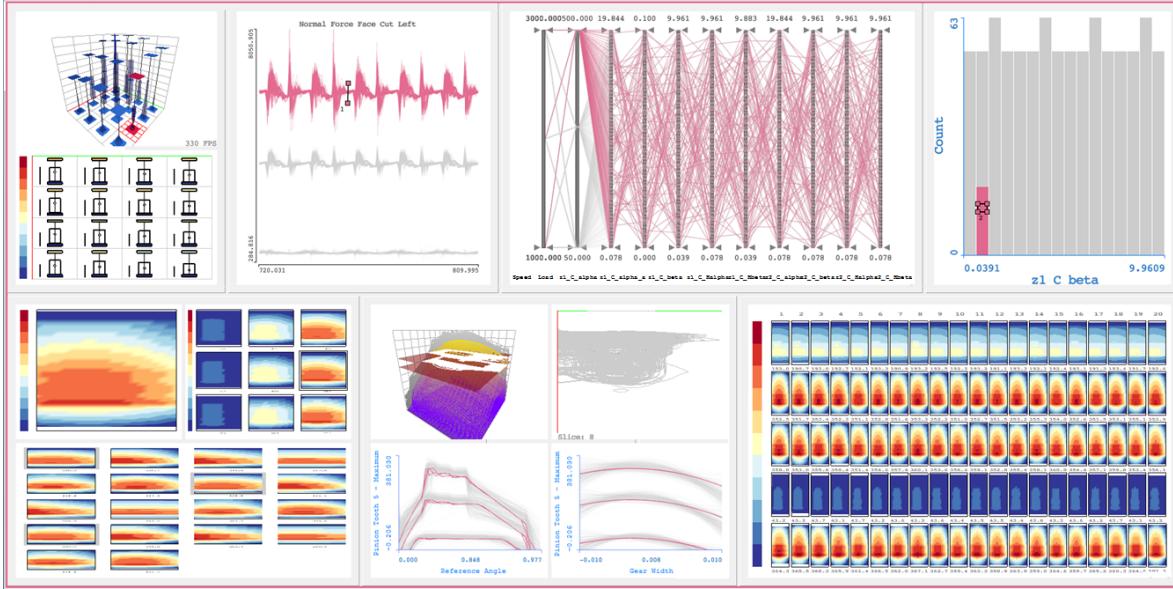


Figure 3.6: An exemplary configuration of the analysis tool. Standard and surface specific views are combined to a CMV system to support analysis of surface ensembles.

and correlations. A suitable way of representing a surface ensemble for that purpose is the aggregation across ensemble members using statistical values. They are depicted in three-dimensional box plots, which are uniformly distributed across the surface domain. Dependent on the number of box plots used, the three-dimensional view can be cluttered. Therefore, once the analyst detects an interesting scope, she can zoom in by using two-dimensional box plots in separate view. These box plots illustrate the statistical distribution at a higher resolution, while at the same time keeping notion of the current position in three-dimensional view. In this way, two-dimensional box plots offer more detailed insights into the distribution of surface values in the specified area.

At some point of the analysis, the user might want to investigate and compare instances of surfaces themselves. A hierarchical arrangement of heat maps, representing a set of data surfaces, supports such tasks by providing three levels of detail: a context overview, a representation for brushed surfaces, and a display of an individual surface. Surface aggregation profiles, which are extended by intersecting contours of surfaces with a horizontal plane, further support the perception of data surfaces at this level. However, visualizing original data surfaces as three-dimensional surfaces at the bottom level is essential and highly helpful for a deeper understanding of the named representations as well as the data surface itself.

In combination with the surface specific views, which are in detail described in the following sections, we also include standard views, such as histogram, scatter plot, parallel coordinates, and others, for analysis. Altogether, they are integrated into a coordinated multiple views system, forming a powerful collection of techniques for interactive visual analysis of surface ensembles (Figure 3.6).

View	Detail			Scale		
	Low	Medium	High	Low	Medium	High
3D Box Plot		✓				✓
Heat Map - Context		✓				✓
Contours - 2 Planes			✓			✓
Contours - 1 Plane			✓			✓
2D Box Plot			✓			✓
Heat Map - Brushed			✓			✓
Heat Map - Detail				✓	✓	
3D Surface				✓	✓	

Table 3.3: Classification of newly proposed views for analysis of surface ensembles. Constant switching between views supports the trade-off between detail and scalability.

During analysis, the analyst constantly finds herself faced with a trade-off between detail and scalability: each of the surface specific views supports different analysis tasks at various levels of detail and no view alone is sufficient for a thorough exploration and investigation of surface ensembles. Table 3.3 shows a classification of the presented surface specific views with respect to detail and scalability. As can be seen, some views provide a better overview of a large number of surfaces, while others support a more detailed analysis, but do not scale well.

Multi-resolution box plots display summaries of distribution characteristics of a large number of surfaces, but only show a surface ensemble as aggregation across members. In contrast to that, intersection contours are obtained from individual ensemble members, but due to high scalability, they do not provide many details on single surfaces. Heat maps display individual surfaces in more detail as they color-code function values across the entire domain. However, we can only show a limited number of ensemble members simultaneously. When choosing suitable views for analysis, details can be sacrificed to

show a large number of ensemble members, while the size of a surface ensemble can be reduced for the benefit of further details. The proposed views support constant switching within those limits, which offers a powerful opportunity for drill-down.

The methodology proposed in this thesis consists of various contributions that we make concerning the interactive visual analysis of surface ensembles. We extend the approach presented by Matković et al. [MGKH09] by developing novel aggregation techniques as well as building upon existing views such as aggregated surface profiles and the three-dimensional display of surfaces as such.

We describe the functionality of these views by means of the lighting example, which was introduced in the previous section. As a starting point, we consider the collection of light distributions upon a desk surface, which result from measuring illuminance values under 576 different lighting conditions, as a surface ensemble. We perform an interactive visual analysis of these surfaces in order to identify lighting settings that ensure a proper illumination of the desk area. This includes adequate illuminance values of at least 100 lux as well as an even and neutral colored lighting.

3.3.1 Three-dimensional Box Plots

At the beginning of interactive visual analysis, the analyst is interested in obtaining an overview of light distributions as well as in identifying overall patterns that can be observed throughout the set of lighting settings. At this stage of analysis, it is essential to concisely summarize important characteristics of the data set's distribution. We contribute to these issues by providing aggregation across ensemble members at a certain number of locations within the desk surface. As Potter states in her survey about methods for presentation of statistical information [Pot06], descriptive statistics can be used to efficiently describe characteristics of a distribution of values in a simplified way. Out of a large number of methods for visual presentation of statistics, box plots have become a standard technique for presentation of data summary that consists of five statistical measures, namely minimum, maximum, median, lower quartile, and upper quartile. Box plots furthermore enable a more straightforward comparison between multiple data sets [Pot06]. As our surfaces are defined in a three-dimensional domain, we take the standard box plot to a higher level by using three-dimensional box plots for visualization of aggregated light distributions, as illustrated in Figure 3.7.

When analyzing light distributions over a desk area, being able to draw conclusions about the illumination of specific parts of the area is very important for evaluation. It is therefore helpful to view local characteristics for parts of the desk area, instead of investigating overall statistics, which apply to the entire area. For this purpose, the desk surface is divided into a certain number of sub-domains, we call them *cells*. Per default, analysis starts with two times two quadrants (Figure 3.7, left), because this does not stress the analyst's perception, while at the same time depicting a minimum of spatial differences. The number of subdivisions in *x*- and *y*-direction can be chosen independently so that they meet the analyst's needs. For each cell, we collect the

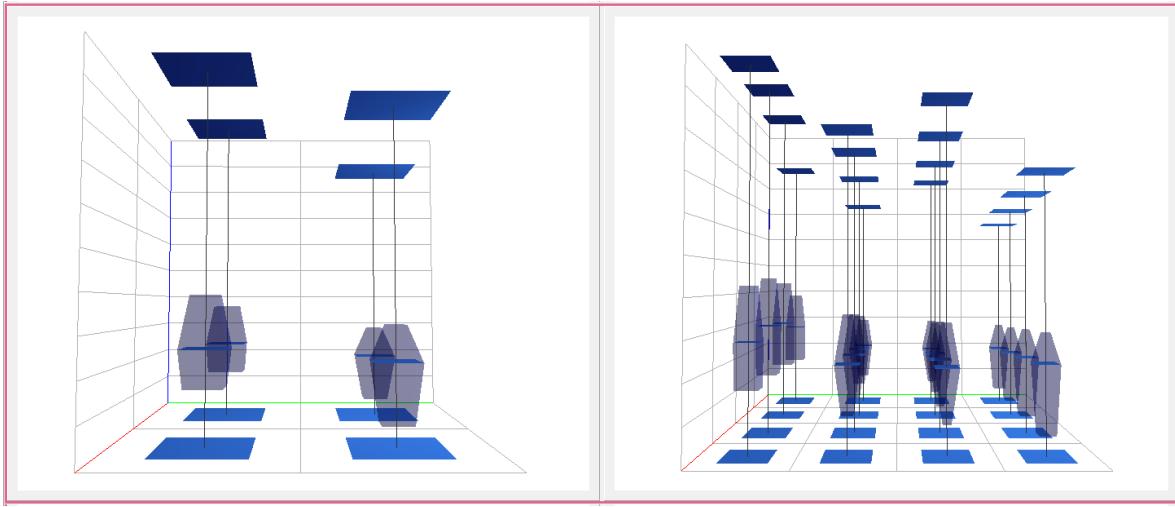


Figure 3.7: Statistical aggregation across 576 light distributions. The same surface ensemble is represented by four (left) and 16 (right) three-dimensional box plots.

illuminance values that are contained within the cell's area across all light distributions. This yields one list of values per cell, from which we compute the descriptive statistical measures: minimum, maximum, median, and quartiles.

Based on these quantities, we construct a three-dimensional box plot, which is based on the typical construction of a two-dimensional box plot, as representation of one cell. Each box plot is drawn at the center of the cell it represents. Two planes indicate minimum and maximum of illuminance values that arose at this part of the desk surface across all lighting setups. The size of a single box plot, and therefore the width of the planes, can be arbitrarily chosen within certain limits, which emerge from the total number of box plots that are rendered.

Usually, in two-dimensional space, the interquartile range of a box plot is represented by a box. Taking this representation to three-dimensional space we would obtain a cuboid as representation. However, we intend to add simplified histogram information to our three-dimensional box plots. Therefore, the interquartile range, which is the area between upper and lower quartile and includes 50% of the values, is visualized by approximating the distribution of the values it contains. For this purpose, we measure the frequency of values at the median and the two quartiles by assuming intervals with a certain tolerance around these quantities and then counting how many elements fall into each interval. The width of the cuboid at lower quartile, median, and upper quartile is then modified to be proportional to the calculated frequency at these locations. This changes the original cuboid into a polyhedron that represents the distribution of values within the interquartile range. The representation described here is inspired by the so-called histplot, which was first introduced by Benjamini [Ben88].

The polyhedron is intersected by a third plane, which is drawn at the median value and whose width is also proportional to the frequency at this location.

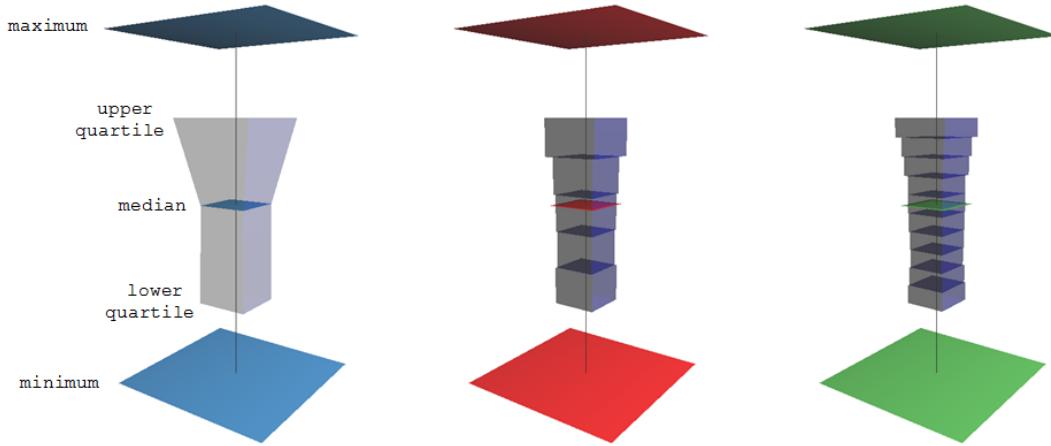


Figure 3.8: Box plot in different interaction states: None with simplified distribution (left), Hovered with histogram (middle) and Permanent with refined histogram (right).

As an example, we investigate the light distributions resulting from 576 different lighting conditions as described in the illustrative example (see Section 3.2). The box plots representing the left-hand quadrants in Figure 3.7, left, show higher maximal illuminance. At a high level of abstraction, this indicates lighting conditions that cause extremely high illuminance values at the left half of the desk surface. When additionally considering a higher resolution of 16 three-dimensional box plots (Figure 3.7, right), we can limit this conclusion to the leftmost cells. Looking back at the given scene (Figure 3.3), we can see that this is exactly the area of the measurement surface that faces the window. We can observe alternating high and lower maximum illuminance values, which were not visible at a lower resolution of four quadrants. This is a consequence of blinds, which cause shadow stripes on the desk, as can be seen in Figure 3.5, c and f.

Interactive visual analysis intends to include the analyst with her capabilities in the analysis process, which is often realized by providing user interaction. Enabling the analyst to interact with the displayed box plots allows her to investigate specific parts of the desk surface, which she finds interesting. Throughout analysis, a three-dimensional box plot can have one of three states, which are displayed in Figure 3.8: *None*, *Hovered*, and *Permanent*. They are triggered by different user inputs and have various impacts on the appearance of box plots and on-demand details.

Per default, all box plots have the state *None*. That means no interaction has taken place yet, as we are at the beginning of the analysis. Box plots are displayed with a simplified histogram, as described above (Figure 3.8, left). They are useful, when analysts are interested in getting an overview of the global range and behavior of illuminance values under different lighting conditions. They also provide guidance for further proceedings, which depend on the analysis tasks and application domain.

After having obtained a basic idea of the light distributions, the analyst might need to get a better insight into the underlying data. This can be used in order to perform a strategic investigation of regions of interest, for example the part of the desk surface that is directly in front of the user, with the aim of solving the given analysis tasks.

The analyst can request a box plot to be visualized in more detail by moving the mouse pointer over it. This changes the box plot's state to *Hovered* and with it also its appearance. The simple approach for displaying frequency information using a polyhedron is refined in such a way, that the same distribution is now represented in more detail by using a method similar to a three-dimensional histogram. The interquartile range is divided into a certain number of non-overlapping, equally sized bins, for which frequency is calculated. The number of bins, and therefore the degree of refinement, can be arbitrarily chosen by the analyst. Individual histogram bins are then represented by using a cuboid, which is drawn at the position of the value interval it stands for and whose width is again proportional to the respective frequency. Figure 3.8, middle, depicts a box plot in hovered state. When comparing to the simple approach for distribution, one can see that the histogram provides more information about distribution of values, even when consulting only a relatively low number of bins.

We provide another level of detail on mouse-over by subdividing the cell of the hovered three-dimensional box plot and additionally presenting two-dimensional box plots in a separate view for each of the subdivisions. Two-dimensional box plots and the information they present are in detail described in the following section. However, as the mouse pointer leaves the box plot its state is set back to *None*, with the consequence that frequency information within interquartile range is again represented by a polyhedron and two-dimensional box plots are hidden.

Requesting a more detailed representation by hovering over three-dimensional box plots efficiently allows for a quick inspection of overall tendencies. If the analyst intends to further investigate one specific part of the desk surface, she can select the corresponding three-dimensional box plot permanently. As a consequence, the histogram is used again for displaying the frequencies of values (Figure 3.8, right) and two-dimensional box plots are also shown, just like we know it from *Hovered* state. But in contrast to *Hovered* state, the details on-demand do not vanish when the mouse pointer is moved out of the element. This ensures, on the one hand, that zooming and panning can be performed without loosing details and, on the other hand, that mouse-over actions can be executed on two-dimensional box plots, as will be explained in the following section. In this way, the provided additional information for the selected box plot can be further investigated using interaction techniques. To switch back to a lower level of detail, a three-dimensional box plot in *Permanent* state can be reset to *None*.

For an effective analysis of inter-dependencies between control parameters and resulting lighting distributions, e.g. the impact of halfway opened blinds on the illuminance values, it is essential to be able to select a subset of the given data, which is then investigated using various views. Actively brushing individual light distributions using this view is not provided, but the analyst can focus on a certain subset of data by brushing desired parameters in another view, for example in histogram. The three-dimensional box plots then automatically update according to the new brush. The computation of statistical measures that are represented is basically the same as before, except for the fact that only the values of brushed light distributions are taken into account.

At the same time following a focus and context approach, we also provide box plots

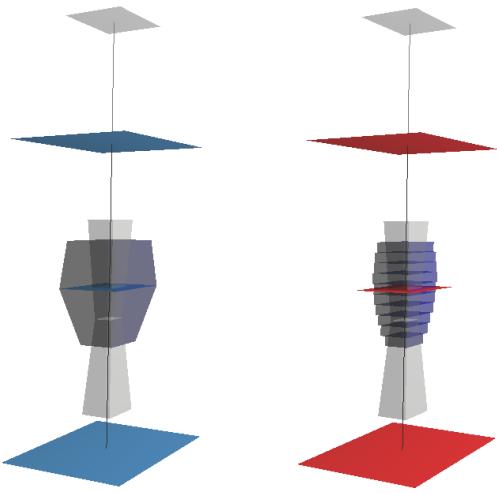


Figure 3.9: 3D box plot: brushed information are displayed in color, while context is given in grey (left). The same box plot when hovered (right).

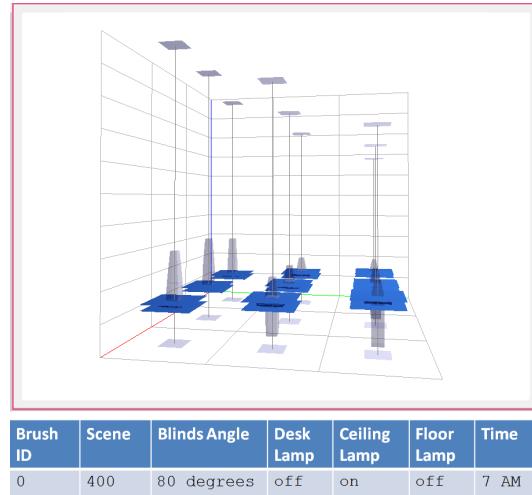


Figure 3.10: A brush resulting in evenly distributed illuminance (top). The data view reveals the control parameters' exact values (bottom).

representing the context data, i.e. the values of light distributions for all setups. Due to their representative nature, user interaction has no effect on context box plots (Figure 3.9, right), while interacting with brush box plots effects their state as described above. To not attract too much of the users attention, context box plots are drawn smaller and in light gray, while the more important box plots that show the brushed values and therefore the analyst's focus, remain colored (Figure 3.9, left). This allows for establishing relations between brushed data in focus and the entire data set.

Using three-dimensional box plots together with brushing also enhances the identification of lighting settings that produce an even distribution of light at reasonable illuminance values. Because three-dimensional box plots clearly show the overall distribution of illuminance values of a brushed subset of data, the analyst can efficiently and quickly determine if the resulting light distributions are in principle acceptable or not. Three-dimensional box plots in Figure 3.10, top, show a brush that results in evenly distributed light. In contrast to what is desired, the overall light provides only relatively low illuminance values. Additionally, from the context box plots we can obtain the information that there are lighting conditions causing extremely high illuminance values that can be interpreted as spot lights. The data view, which shows the brushed control parameter settings (Figure 3.10, bottom), reveals that the selected light distribution originates from simulation with daylight at 7 am, nearly closed blinds at an angle of 80 degrees, desk and floor lamp being switched off, while ceiling lamp is switched on. The corresponding scene is depicted in Figure 3.5a.

In many application domains, in particular in the context of gearing systems, which are described in Chapter 4, locations with a high variation of values, e.g. a large difference between minimum and maximum value, are of special interest, because they

indicate extrema and fluctuations. For supporting the identification of such samples with striking characteristics, we provide an interface for interactively filtering the box plots according to their variation. A slider can be used to set a variation threshold, which has to be exceeded by a three-dimensional box plot in order to be rendered. In general, the possible values for the variation threshold range from the overall minimum difference between the box plots' minimum and maximum values to the overall maximum difference. For simplification purposes, we consider the three-dimensional coordinate system, which uses normalized coordinates ranging from -1 to 1 on all three axes, for slider range. As the minimum value of a box plot cannot be less than -1 and the maximum value can be 1 at most (this is determined by the range of the z-axis), we assume the variation threshold to be between 0.0 and 2.0.

3.3.2 Two-dimensional Box Plots

Three-dimensional box plots, as we described them in the previous section, are suitable for getting an overview of the overall distribution of illuminance values, but do not provide enough detailed information for in-depth analysis of a single cell. Even when heavily increasing the number of subdivisions along both axes, we cannot perform a detailed analysis, as the perception of details and single box plots is disturbed by occlusion and distraction by surrounding objects. Therefore, we additionally propose another level of detail in the form of two-dimensional box plots, which are triggered by selecting a three-dimensional box plot.

These box plots are presented in a separate view (Figure 3.11, bottom) that contains two axes, which correspond to the x - and y -axis of the three-dimensional box plot view. This establishes a spatial connection between the views and eases orientation. Previous selection of a three-dimensional box plot determines the part of the desk surface, which is to be further investigated. Two-dimensional box plots provide additional information on the lighting distributions within that part.

The area is further subdivided, resulting in each subsection being represented by one two-dimensional box plot. The underlying statistical measures are computed in the same way as before by only considering the values that belong to the specific subsection. Again, the number of subdivisions in both directions can be chosen by the analyst. The two-dimensional box plots are arranged in a matrix that reflects the positions of corresponding subsections within the coordinate system of three-dimensional box plot view. Accordingly, the covered subsections are highlighted in the forenamed coordinate system to support the analyst's perception of spatial relations (Figure 3.11, top).

Three-dimensional box plots show decreasing maximum illuminance values from the left to the right side of the measurement surface (Figure 3.11, top). As mentioned in the previous section, high illuminance values on the left-hand side result from the incidence of light coming through the window. One could now conclude that light intensity decreases with increasing distance from the window, but when additionally considering two-dimensional box plots we can see that the light distribution in fact behaves slightly different (Figure 3.11, bottom). Colors of the two-dimensional box plots corresponding

to the selected three-dimensional box plot (Figure 3.11, top) indicate higher illuminance values to the right of the 3D box plot's sub-domain (Figure 3.11, bottom). This contrary observation can be explained by shadow stripes on the desk surface, which result from the blinds being half open. In this way, additional information that are provided by two-dimensional box plots at a higher level of detail offer enlarged insights into the distribution of illuminance values within a certain sub-domain of the desk surface.

The natural starting point when thinking about a suitable approach to realization of two-dimensional box plots, is to use a common scale for all of them to ensure comparability. As box plots are arranged in a matrix, only elements within one row or one column of the matrix can be compared side-by-side. As a consequence, we decided to use a common scale per matrix row. The range of each scale needs to cover all statistical measures that are represented by two-dimensional box plots of the corresponding row. This means that outliers of the statistics, for example extremely high illuminance values caused by specular highlights, heavily stretch the scale's range although most of the box plots only cover a part of that scale. As a consequence, the majority of box plots is displayed in a very small size, which significantly complicates the perception of statistical values. Therefore, a common scale per row turned out not to be the right choice for presenting the two-dimensional box plots.

For optimal exploitation of the screen space and for simplification of box plot investigation, we thus decided that the two-dimensional box plots should all be of the same size. In this case, we need another way to depict the actual values of statistical measures. Rectangles represent the minimum and maximum of a box plot, while a circle indicates the median value. We color code these shapes using colors that range from dark blue to yellow to dark red, based on their value. As we use a discrete color scale representing 11 data classes, which is drawn at one side of the view for reference, the color code itself only allows for a rough classification of the actual values, which should be sufficient for gaining a quick overview. The interquartile range is marked using a box, where the lower and upper sides respectively stand for the lower and upper quartile, just as we know it from commonly used box plots.

Figure 3.12 depicts an example of a regular two-dimensional box plot. As the design of two-dimensional box plots and the color coding do not provide for exact values that may be needed at some point of the analysis, we allow for user interaction in the form of mouseover. Hovering over one of the shapes representing minimum, median, and maximum as well as over the upper or lower side of the interquartile box has two effects: (1) A point on the color scale indicates the position of the exact value of the hovered measure, and (2) The exact value of the statistical measure is displayed next to the hovered shape. In this way, we do not provide too many information right from the beginning, but allow for request of exact values, if needed.

When using two-dimensional box plots in the context of searching for an appropriate illumination, it is essential that the analyst can quickly perceive if an illuminance value is relatively high or low. This implies that she knows the global range, in which the remaining illuminance values are located. When dealing with equally sized box plots, the user cannot assess the overall range covered by a single two-dimensional box plot at

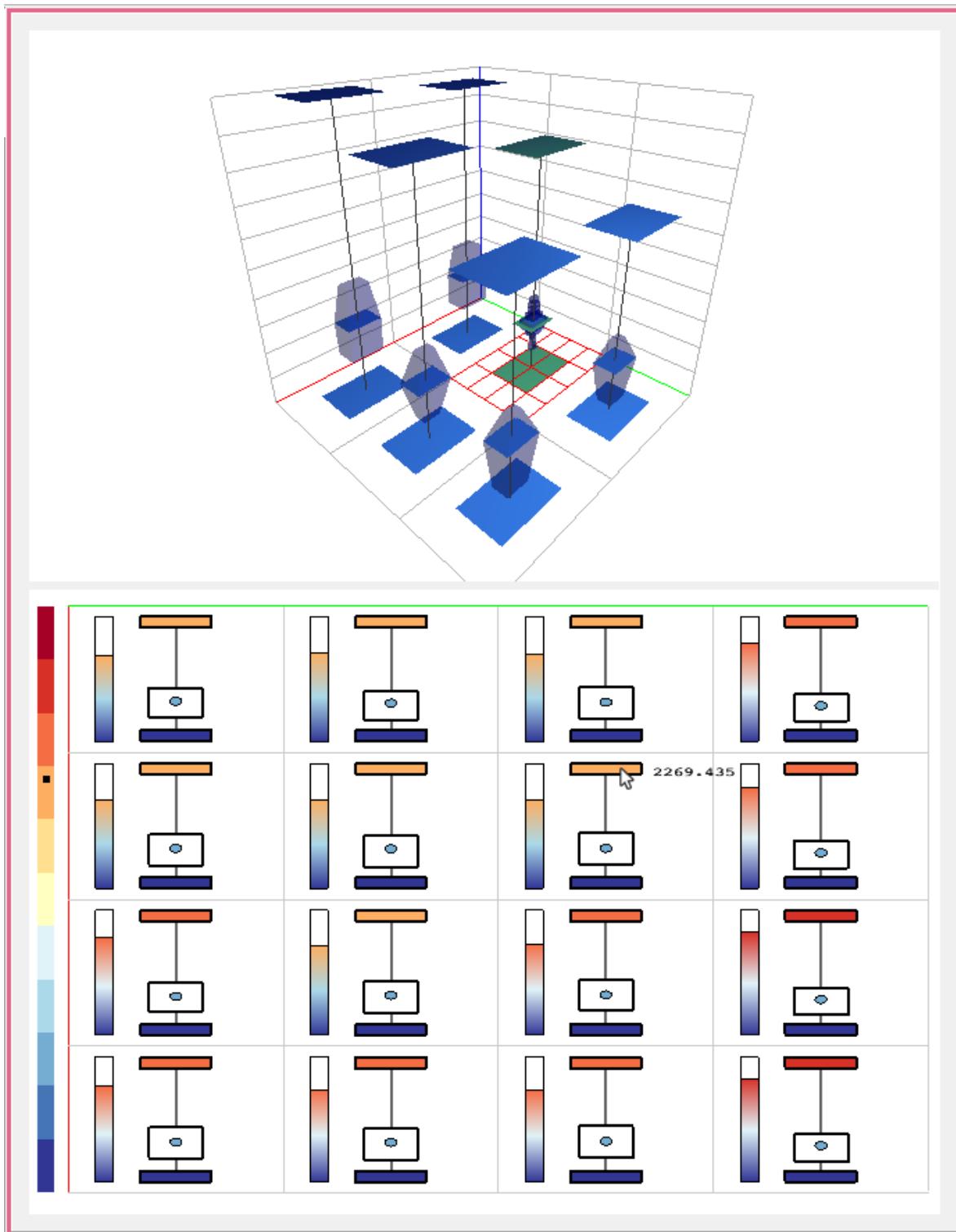


Figure 3.11: 2D box plots (bottom) displaying details for permanently selected box plot (top). Corresponding sub-divisions of the surface domain are marked. Mouseover displays exact median value and marks it on color scale (left).

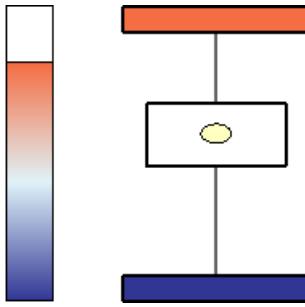


Figure 3.12: 2D box plot: colors encode minimum, median, maximum. A bar (left) indicates the box plot's range.

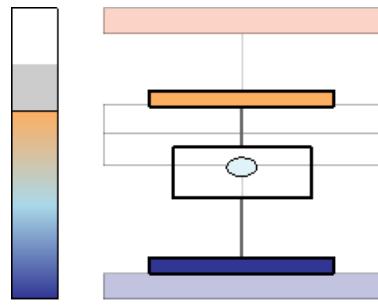


Figure 3.13: The same box plot with brush. Brushed values are displayed relative to the context's range.

first sight. She could take the positions of minimum and maximum values on the color scale into account, but that would involve additional cognitive stress. Also, this would not allow for comparison of multiple two-dimensional box plots. To not lose the information about a box plot's overall range that a common scale would have provided, we additionally offer a rectangular range indicator next to the box plot (Figure 3.12, left). Within a black frame, which represents the global range, a colored rectangle indicates the part of the global range that is covered by the box plot. Top and bottom color of the rectangle correspond to minimum and maximum values, which ensures consistency with the colors used for the related box plot. A color gradient is used to additionally enhance the perception of the covered range.

The two-dimensional box plot view is linked to all other views and automatically updates when a brush is added or changed. As is the case with three-dimensional box plots, actively brushing several light distributions by means of two-dimensional box plots is not supported. However, if an interesting subset of data is brushed in another view, statistical measures are recomputed by considering only brushed data. The original box plots, which represent the entire data set, are kept as context information. They are depicted with more transparent colors as well as smaller lines.

Box plots that represent the brushed data remain colored. They are drawn with a slightly smaller width to not occlude the context box plots in the background, but with regular line width and opaque colors. Figure 3.13 shows such a two-dimensional box plot representing brushed information. To be able to directly compare a brush box plot with its corresponding context, the statistical values of the brush box plot are depicted relative to the context box plot's range. Accordingly, the global range indicator is updated. The colored rectangle depicting the range of brushed values is drawn relative to a grey rectangle, which is added to represent the range covered by the context box plot. Color coding and mouseover interaction function as described for regular box plots.

3.3.3 Heat Maps

Data that are structured as surface ensembles often require an analysis in greater depth. For analyzing patterns and interrelations among the light distributions resulting from

different setups, a side-by-side comparison of individual distributions is essential. To perform this task, there is a need for relatively small representations for each surface. Heat maps are suitable for this purpose, because they encode illuminance values across the desk surface and striking colors can draw the visual attention to potentially critical regions with extreme values.

For each sample point of the surface, the corresponding illuminance value is mapped to one of the colors of a discrete, diverging color scale. Choosing an appropriate scale was supported by the online tool ColorBrewer.org [HB03], that was originally developed for selection of color schemes for maps, but offers various useful features that also apply to our application domain. The chosen color scale represents eleven data classes using colors that range from dark blue for low values to yellow to dark red for extremely high values. If desired, for example when experts of a specific application domain are used to certain colors, a custom color scale can also be applied. The color values between the sample points result from interpolation of sample point colors. When providing color-coded illuminance values, it is essential that analysts are able to easily interpret the colors' meaning. For this purpose, the color scale is drawn as reference on the left side of the heat map arrangement.

Heat maps have to be of a certain minimum size to be able to identify regions of interest. Due to this issue, we can only depict a limited number of them simultaneously. We therefore suggest to use heat maps when data analysis has already progressed and the number of data points has been reduced to a reasonable amount using brushes.

Analysts want to gain insights into overall characteristics of the data, but they also need to investigate individual data surfaces that they find interesting in more detail. Comparing different sets of ensemble members is also important for analysis. A hierarchical approach supports such procedures by enabling an investigation of surface ensembles at three different levels of detail. We provide a context display at the top level, one display for brushed surfaces at a mid-level, and a detailed one displaying a manually selected surface at the bottom level (Figure 3.14). In all displays the analyst can use zooming and panning for temporarily switching between focused and contextual representations within one display. Furthermore, for efficient classification a scalar aggregate, in our case the maximal illuminance value, is depicted for each light distribution.

The context display can be used for specifying a reference subset of light distributions by setting a current brush as context information (Figure 3.14, bottom). All corresponding heat maps are displayed in the form of a matrix, one heat map for each brushed simulation run, but for a fixed surface attribute, which was previously chosen. In case of the lighting example, we have only one such attribute, i.e. the light distribution on the desk area. The reference subset is not automatically updated when the brush by which it was defined changes. Instead, it stays the same until another brush is manually determined to be used as context information from then on. In this way, the analyst is enabled to perform a side-by-side comparison of two different subsets of a surface ensemble or a comparison of the context brush with a refinement of such.

The mid-level display shows heat maps for the light distributions that are currently brushed (Figure 3.14, top right). When the brush changes the display updates auto-

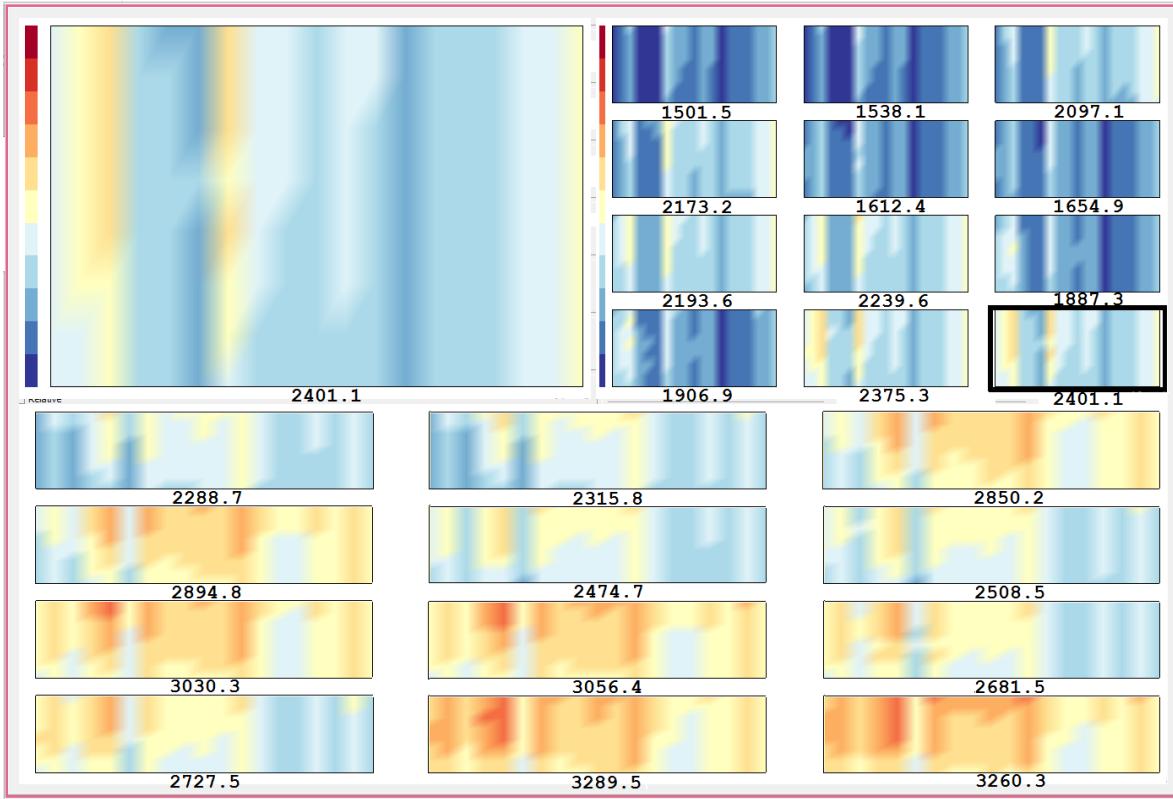


Figure 3.14: Hierarchical heat map view for a selected surface attribute. Brushes can be set as context (bottom). Currently brushed data points (top right) are displayed next to a detail display (top left). All displays support zooming and panning.

matically. This display allows for investigation of a selected subset of light distributions in a larger size, which provides more details. Heat maps are arranged in the same way as in context display. Together with the context, the mid-level display can be used for comparison. At this level, the analyst can also manually select one of the heat maps via mouse click for further detail analysis.

The corresponding surface is then shown in the detail display, which forms the bottom level of the hierarchical heat map arrangement. The display only shows one heat map at a time, drawing the analyst's attention solely to the selected light distribution (Figure 3.14, top left). By this means, a precise examination of the distribution of illuminance values resulting from a specific lighting setting is enabled.

In Figure 3.14, the hierarchical heat map approach is used to compare light distributions that result from different opening degrees of blinds. Distributions that emerge from blinds being opened by 35 degrees, where zero degrees means horizontal slats, are set as context surfaces (Figure 3.14, bottom). As an example, Figure 3.5e shows a simulated illumination of the workplace, which also results from blinds with an opening of 35 degrees. Surfaces resulting from blinds with an opening of 10 degrees are depicted in brush display for comparison (Figure 3.14, top right). All of the light distributions were measured at midday (1 p.m.), when the sun provides a lot of light. One of the

distributions with an opening of 10 degrees and higher illuminance values was chosen for detailed analysis (Figure 3.14, top left). In general, we can see that significantly more light falls on the desk surface when blinds are halfway open (35 degrees, Figure 3.14, bottom). Context and brush display furthermore depict varying intensities within the light distributions for a fixed opening degree. Such variations may be caused by different values of other input parameters, for example the ceiling lamp being switched on or off. In all displays, but in particular in detail display, we can also observe vertical stripes due to shadows that are cast by individual slats.

For feature-driven analysis it is important to enable the creation of a mental connection between the focus and the context. It is essential for analysts to constantly know from which point of the context the focus emerged. This approach is supported by linking all heat map displays. Surfaces that were selected for analysis at a higher level of detail are highlighted in the display showing the current level. Light distributions from the set of context surfaces that are also contained in current brush are highlighted with a light gray underlay in context display. In a similar way, the light distribution selected for detailed analysis is highlighted in brushed display, by replacing the gray underlay by a black frame surrounding the selected heat map in brushed display.

The features that were described above refer to the attribute-centric approach as it was defined in Chapter 1. However, some analysis tasks require to view the data from a different perspective, i.e. multiple surface attributes for a given simulation run. In this view, the run-centric approach (for definition see Section 3.1) is treated differently from that, although the underlying data structures are equal. Again, heat maps are arranged in a matrix, but in a slightly different way: all surface attributes belonging to a certain simulation run are depicted as heat maps in the same row. An empirically determined threshold controls the number of simulation runs – and therefore the number of rows – that can be displayed at the same time. The run-centric heat map view enables the detailed investigation of individual simulation runs by observing all surface attributes belonging to the same run. This allows for determination of overall patterns that become visible across the attributes within one run. It therefore opens up a new perspective for analysis of surface ensembles.

3.3.4 Intersection Contours

Box plots and heat maps, as described above, are suitable techniques for exploring overall characteristics and identifying outliers. Unfortunately, both of them lack scalability, which is why interactive visual analysis requires another view that allows for displaying a larger number of light distributions simultaneously. In many cases, a more in-depth analysis of complex data sets is required in order to achieve a complete understanding of the underlying system. When it comes to investigation of a more exact developing of values regarding one dimension, box plots and heat maps are not sufficient anymore. In addition, at some point of the analysis, analysts want to get an idea of the actual shape of surfaces, which these techniques do not support.

Matković et al. propose to use aggregated surface profiles for exploration of the behavior of surface parameters with respect to one of the independent variables [MGKH09].

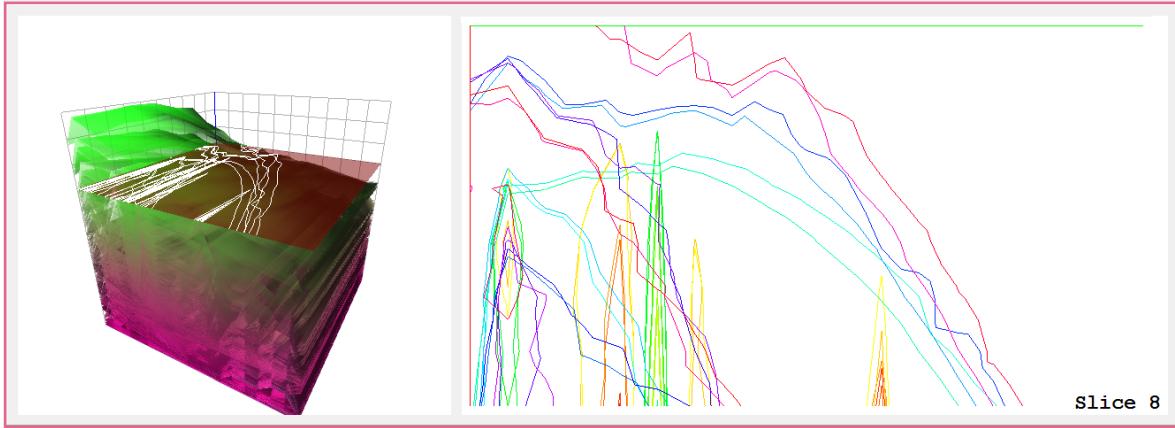


Figure 3.15: *Intersection contours with a horizontal plane displayed in 3D surface view (left) and in a separate view (right). Varying colors help to distinguish contours.*

The representative curves are based on a collection of intersections that are created using a cutting plane parallel to one axis, which depends on the selected variable. Using curves for mid-level aggregation of surfaces allows for better scalability, as a large number (thousands) of data points can easily be depicted at a time. However, it requires a significant amount of cognitive effort to establish a mental model of the surfaces' shape from projections. We therefore extend their approach by intersection contours along the vertical axis, similar to a top view, and therefore open up an additional perspective. There are also analysis tasks, for which analysts need to evaluate the distribution of surfaces in a specific range, for example if a certain minimum intensity of light must be achieved for proper illumination. For this purpose, we add a horizontally aligned cutting plane to the 3D surface display (which is in detail described in the following section), which can be placed at a certain height within the range of the vertical axis. For undisturbed investigation of light distributions within the 3D surface display, the plane can also be hidden, resulting in the intersection contours being deactivated. If active, for every light distribution we compute contours that arise from intersection with the cutting plane. Note, that, depending on their shape, some surfaces do not necessarily intersect with the cutting plane, while others result in more than one contour. The intersection contours are then presented to the analyst in two different ways: (1) they are drawn onto the cutting plane in 3D surface display (Figure 3.15, left), (2) in a separate two-dimensional view (Figure 3.15, right). In this way, we preserve the spatial context information in 3D surface display, while being able to perform a more precise examination of the intersections in 2D contour display.

For differentiation of contours in two-dimensional view, we use different colors for contours that belong to different surfaces. Depending on the number of surfaces, we propose an equidistant sampling of hue in HSV color space. This yields an individual color for each set of contours representing the same surface, where each set can contain none, one, or multiple contours (Figure 3.15, right). If brushed light distributions exist, the corresponding contours are drawn with the respective brush color, while context con-

tours are colored in light gray. Analysts might want to closer investigate certain regions of interests in contour view. Zooming and panning allow for a detailed investigation of contour shapes, guided by the analysts' findings.

Choosing appropriate lighting conditions for a workplace, like the desk in our illustrative example, mainly relates to maintaining a specified minimum level of light intensity. In Figure 3.15, left, the vertical position of the cutting plane is equivalent to a relatively high illuminance value. Consequently, the contours on the right represent 31 light distributions that contain such high illuminance values. The intersection contours themselves exactly show where the mentioned values arise on the desk area. We can see that, for lighting settings that produce such high illuminance values, the distribution is rather irregular as high values rather emerge at the left side of the desk (Figure 3.15, right). This conveys that incoming light from the window, which is positioned at the left wall, seems to have a significant impact on the measured illuminance on the desk surface. That leads to the conclusion that such irregularities have to be compensated to achieve an even distribution of light.

Furthermore, two clusters of light distributions can be identified from two-dimensional contour view (Figure 3.15, right). We can see peaks emerging from the bottom of the view at certain intervals. They indicate shadow stripes and therefore represent one cluster of light distributions, which probably results from lighting settings with halfway opened blinds. In contrast to that, we can also observe rather round contours, which run from the left upper corner to the bottom of the contour view. They represent a cluster of light distributions that do not contain shadow stripes, which leads to the conclusion that these distributions result from blinds being completely raised. The shape of the corresponding contours may arise from the desk lamp's light beam. By taking a look at the data view, which displays the values of all attributes for selected light distributions, these hypotheses are verified: the data view reveals that 13 of 31 light distributions were simulated with blinds being open at 35 or 60 degrees, while blinds were raised for simulation of the remaining distributions.

At any time during analysis, when she comes across outstanding light distributions, the analyst can create a brush set using the cutting plane, even as part of a composite brush. This selects all surfaces that currently intersect the cutting plane. When the plane is moved, brushes update automatically according to the intersections with the newly positioned plane. Deactivating the mode leads to removal of the brushes that were created using intersections. All representations are linked in such a way that brushing in one view is automatically reflected in all other views.

Because analysts sometimes need to be more precise, we propose a second cutting plane to be used together with the original one for refinement of brushes.

Figure 3.16 shows the eight data surfaces that correspond to the selected scenes from lighting example, which are depicted in Figure 3.5. When both planes are used as brush only those surfaces that intersect with the lower plane, but not with the upper plane are selected. These are exactly the surfaces that have their maximum peak in between the two cutting planes. The brushed data surface in Figure 3.16, top left, represents the illumination depicted in Figure 3.5h. The cutting planes are placed

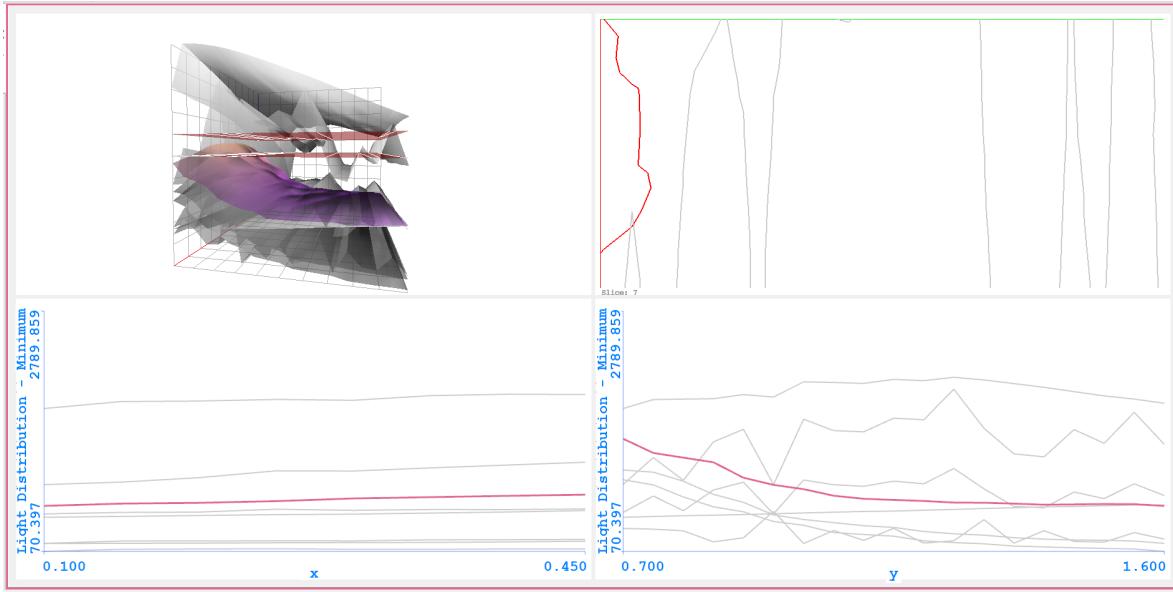


Figure 3.16: Contour view: cutting planes used as brush result in surfaces with a peak in between being selected (top left). Corresponding contours (top right) and surface profiles (bottom) are highlighted.

in such a way that lighting settings resulting in medium illuminance for at least some parts of the measurement surface are selected. Light distributions with too high and too low values are precluded from the outset. Corresponding surface profiles (Figure 3.16, bottom) and contours (Figure 3.16, top right) are highlighted. As before, the analyst can deactivate brushing using cutting planes. When still both planes are shown, the two-dimensional contour view displays the intersection contours of only those surfaces that would otherwise be brushed. However, the analyst can return to only one cutting plane for computation of contours at any point of the analysis.

By scrolling through the slices along the vertical z -axis, the analyst can gain a thorough understanding of how surfaces are distributed. Where parts of surfaces are occluded or not well distinguishable due to large and complex clusters of surfaces, we can easily identify the arrangement of surfaces at different heights in two-dimensional contour view. In contrast to box plots and heat maps, contours offer a more detailed representation of surfaces, independent from a possibly complex distribution of surfaces. From analyzing contours the analyst can draw more precise conclusions about the shape of surfaces. Another benefit is their high scalability. We can easily depict a large number of contours simultaneously without stressing the analyst's cognition.

3.3.5 Three-dimensional Surface Display

The aggregation techniques introduced in the sections above offer great potential for the visual exploration and analysis of surface ensembles.

Still, for a thorough understanding of data surfaces it is essential to present them in their original form, as three-dimensional surfaces. It allows for building an image of

the surface's shape in the analysts' mind, which helps them in understanding and interpreting aggregated representations better. Throughout the whole analysis process, such a mental image of a data surface is a constant source of shape information.

Using such a three-dimensional representation, it is highly challenging to visualize more than a few light distributions at a time, due to problems like occlusion, clutter, and a lack of spatial clues [HJ11]. Such issues make it hard for analysts to identify patterns and structure when observing a set of light distributions. Some research has been carried out in the field of single and multiple layered surface visualization to find factors optimizing the perception of surface shape [Bai09].

In their research, Matković et al. use solid shaded meshes for visualization of surfaces in a three-dimensional coordinate system [MGKH09]. However, this approach is limited to a small number of surfaces, because overlapping does not allow for a continuous impression of an individual surface. As a consequence, analysts can hardly decide which of the visible and occluded parts belong to the same surface.

There are various methods to overcome or at least reduce occlusion and clutter issues when visualizing light distributions in three-dimensional space. Shadows and a proper illumination of the scene facilitate shape discrimination, as they provide cues that support the perception of depth. Transparency can be used to reduce occlusion and clutter, but at the same time it furthers ambiguity [HJ11].

To enable a better understanding of volumetric features, we display light distributions as solid shaded polygon meshes using the Phong lighting model with Gouraud interpolation. Two positional light sources with white light located to the right and to the left of the coordinate system allow for illumination of the scene. This provides additional depth cues, which enable a better perception of the surfaces' shape and help resolving ambiguities. Although it simplifies the discrimination of surface shapes, it does not solve the problems of clutter and occlusion, which complicate the analysis.

When depicting many light distributions simultaneously, the observer can rarely view the interior of a surface ensemble, where a lot of surfaces overlap. To improve spatial perception of a large number of light distributions in spite of occlusion, which occurs when rendering solid shaded surfaces, we additionally use transparency. The idea is to render outer surfaces transparent and inner surfaces opaque, in such a way that surfaces (or parts of them) are kept viewable that would otherwise be occluded by surfaces closer to the observer. In this way, the analyst can look through surfaces close to her to investigate the ones that are positioned farther behind. A linear transfer function maps surface points that are closer to the camera to higher transparency, while points farther are rendered increasingly opaque.

Previous to the mapping, we compute two distance measures, d_{min} and d_{max} , which make up the range in which the transfer function applies. Here, distance means the Euclidean distance of a surface point to the camera position. d_{min} represents the distance of the surface point closest to the camera, while d_{max} is the maximal distance between any of the surface points and the camera.

We obtain a surface point's transparency value α by mapping its distance d to $[0,1]$ using $\alpha = \frac{d-d_{min}}{d_{max}-d_{min}}$. As a result, the surface point closest to the observer is rendered fully

transparent, while the farthest surface point is rendered completely opaque. Adding a small offset ensures that the closest surface points are not invisible.

When the data set is large, computing distances for all surface points of every surface in the data set is highly time-consuming. We therefore propose a less CPU-intensive alternative for determination of d_{min} and d_{max} .

To obtain reasonable mapping results, we have to ensure that the distance range $[d_{min}, d_{max}]$ covers all occurring distances of surface points in the data set. On the other hand, it must not be too wide, as differences between mapping results get very small as the range grows. To use the available range optimally, we have to define d_{min} and d_{max} in such a way that the range is as wide as necessary and at the same time as small as possible. Considering the corners of the coordinate system's bounding box for computation of d_{min} and d_{max} , instead of surface points themselves, satisfies both criteria. Here, we have to distinguish between two setups: (1) The camera lies outside of the coordinate system. In this case, we can compute d_{min} and d_{max} considering the bounding box corners in the same way we did when using surface points. (2) The camera is within the coordinate system. Then we set d_{min} to 0 and only compute d_{max} with respect to the corners of the bounding box. This approach reduces the computing effort to a small number of distance computations.

Figure 3.17 depicts 576 light distributions simultaneously by using three different approaches. When being displayed as solid shaded polygon mesh that is combined with lighting and transparency as described above (Figure 3.17, left), the shape of individual surfaces can be guessed up to a certain degree. We can barely observe fluctuations of illuminance values along the horizontal axis, which leads to the conclusion that some surfaces contain a certain number of spikes along this axis. As already described in the previous sections, these spikes result from shadow stripes caused by blinds. Furthermore, the analyst can roughly estimate the scale of the number of displayed light distributions from this view. Figure 3.17, middle, depicts the wireframe mode, which is widely used for surface display. When depicting a large number of light distributions, the wireframe lines merge into one colored area. The observer can get a notion of the distribution of surfaces, but is hardly able to distinguish between individual light distributions. Fluctuating illuminance values cannot be identified as clearly as with solid shaded polygons. The third alternative is to render light distributions as solid shaded polygons but without any lighting or transparency. As Figure 3.17, right, shows, only the surfaces' envelope is visible to the viewer. This limits the perception of surface distribution and also does not allow for an estimation of the number of light distributions that are simultaneously displayed.

3.4 Implementation and System Integration

The presented interactive visual analysis approach has been integrated in a coordinated multiple views system, called ComVis [MFGH08]. The tool allows for rapid prototyping of new visualization techniques by providing views that are flexible in both use and implementation. Other views include standard multivariate visualization techniques,

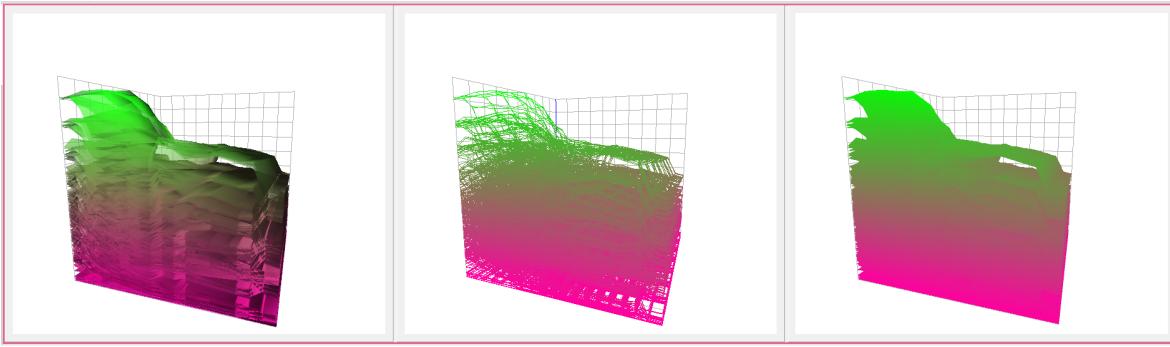


Figure 3.17: 3D surface view: solid shaded polygons with lighting and transparency (left), wireframe mode (middle), and solid shaded polygons without additions (right).

such as histogram, scatter plot, and parallel coordinates, as well as more advanced views like curve view [KMG⁺06] and trajectories view [MGS⁺15]. In addition, the system provides a data view, which supports the analyst with details, by displaying the exact original values of attributes in a table.

Linking between various views is based on brushing several simulation runs. For this purpose, the analyst can choose a view and select a subset of interesting values by defining a single brush or a composite brush, which combines multiple brushes using Boolean operations. The resulting brushed data are immediately displayed in all other views. Visualizations are positioned inside a main window leaving the possibility of choosing an appropriate layout from predefined settings to the user. Saving and loading the current analysis status is also provided.

Integration of the different types of view builds on a data model that highly supports the use of multiple-runs simulation data in such a way that it treats complex data structures as atomic units. Multi-threading accelerates the processing of large surface ensembles and provides good responsiveness of the application.

The visualization techniques presented in this thesis are implemented as new types of view, but also extend existing views. We enlarge the proposed approach of aggregated profiles along one axis by contours that originate from the intersection of surfaces with a horizontal plane. Furthermore, some properties such as lighting and a solid shading mode are added to the surface display to improve the perception of multiple surfaces when observed simultaneously. A new view is established, that allows for statistical aggregation across surface ensemble members, using three-dimensional multi-resolution box plots as well as two-dimensional box plots for detailed inspection of selected data points. We also establish a hierarchical arrangement of heat maps as new view, which supports analysis of surfaces at three levels.

All views are implemented using Qt and C++ as well as OpenGL for rendering.

4. Case Study: Advanced Gear Stress Analysis

In Chapter 3, we describe the proposed interactive visual analysis approach with reference to an illustrative example. As this example was chosen to be easily understandable, it does not represent a wide range of application domains. It therefore cannot be conducive to the evaluation of the presented methodology. This chapter presents a case study, which deals with the analysis of stresses in gear mechanisms.

Gears are of central importance for the design of engines and machines. They have been used for several thousand years to transmit motion [Rad12, Col87] and have become an integral part of mechanical engineering [Dra88]. A wide range of different mechanical devices rely on their performance, such as powertrain systems, wind turbines, and movements of watches, to name just a few. The gear's design affects key functions, such as service life, performance, noise emission, and others. To ensure an appropriate performance for specific applications, gears have to meet high requirements on accuracy and reliability as well as low noise and efficiency. As the application domains rapidly develop further, constant optimization of gear design is essential to keep up with the increasing demands on gear mechanisms.

This means especially avoiding fatigue and failure, since components that do not work properly may cause great damage to the machine. The risk of failure and fatigue has been an early field of research [Shi67] and can be reduced by using suitable material and lubrication or adjusting the load and other parameters, for example. Another way to prevent failure, such as tooth breakage, consists in a regular distribution of forces, which arise when teeth are in contact. This applies to the distribution both between operating gears and along tooth surfaces of mating flank pairs.

To establish a good gear design, we have to understand how gear wheels need to be constructed to achieve low and evenly distributed normal forces along teeth. This is complicated by forces acting in other directions, due to deformation or bending of

bearings and shafts, for example. Such distortions can even lead to a phenomenon called edge load, which occurs in the form of only a corner or edge of a tooth being in contact with another tooth (M. Sopouch, personal communication, December 4, 2015). These phenomena can cause unexpected and undesired working behavior, such as jamming or noise, and a highly uneven distribution of forces [Flo00].

There are too many possible variations of quality characteristics, like material, topography, load, speed, and others, to test for a gear design that achieves the best performance [Han15]. Instead, we want to get a better understanding of their impacts on gear stresses by using the interactive visual analysis approach presented in Chapter 3. The following section describes the components of a gear mechanism and gives an overview of basic functional concepts.

4.1 Components and Working Principles of Gears

The idea of equipping wheels with teeth and using their rotation to transmit forces is simple. Every observer may imagine that one gear drives another one by means of *meshing*, i.e. the engagement of mating gear teeth. To someone who has never studied gears, it might seem that there is no need for further explanation. However, a proper operation of gears relies on a huge amount of geometric theory and in the field of gears theory and practice are very closely linked [Col87]. Therefore, this section provides some theoretical insights into the field of gears.

Fundamentals

A *gear* is a toothed wheel used in many types of machinery to transmit motion from one shaft to another. Transmission is achieved either by a uniformity or by changes in rotational speed, direction, and shaft torque [Dra88]. Various parameters, including mesh geometry, lubrication, load, and speed, determine the characteristics of a gear's motion transmission. In this thesis, we will focus on the most common type of gears: cylindrical gears. In particular, we will deal with involute spur gears and helical gears.

Figure 4.1 presents an illustration of both gear types.

The teeth of *spur gears* are parallel to the gear axis and spur gears are mounted on parallel shafts. For involute gears the tooth profile forms the involute of a circle (Figure 4.1, left). They satisfy the so-called *fundamental law of gear-tooth action*, which states how gear tooth contact has to be carried out in order to achieve a constant angular velocity ratio and therefore ensure a smooth motion transmission [ZFB10]. Nevertheless, each time teeth engage and collide the impact causes loud noises.

In contrast, *helical gears* operate more smoothly and with less noise. The leading edges of their teeth are not parallel to the gear's rotation axis, but are set at an angle to it, which is known as the helical angle (Figure 4.1, right). By letting the helical angle approach zero we reach the spur gear as a special case of the helical gear [Col87]. By mounting them on perpendicular shafts, helical gears can also be used to adjust the rotation angle by 90 degrees.

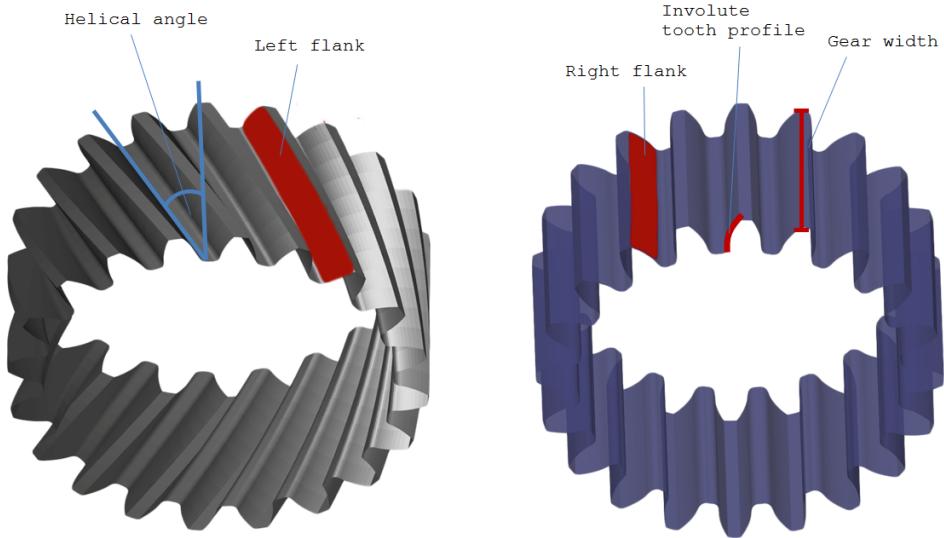


Figure 4.1: Involute spur gear (left) and helical gear (right). Right and left flank as well as gear width, involute tooth profile, and helical angle are marked.

When making up a cross-section of a helical gear perpendicular to its axis we can see that the result is the same as the cross-section of a spur gear [Col87]. Therefore, we can conclude that most concepts that apply to spur gears also apply to helical gears and due to their simplicity, we will consider involute spur gears from now on.

Nomenclature and Kinematics

For the description of gear configuration and establishment of contact, we consider a simple gearing system that contains a pair of involute spur gears. Usually such a gear pair consists of a smaller driving wheel, which is referred to as the *pinion*, and a bigger driven wheel, which is called *gear*. The regions of a tooth that are in contact with its corresponding gear are called *flank*. One tooth has two such flanks commonly referred to as driving and driven flank. However, this association depends on the direction of rotation and the loading torque and may change as the wheels turn in the opposite direction. Due to this issue, we refer to one flank of a tooth as right and the other one as left flank and keep this reference whatever the rotating direction. The *gear width* is measured along the gear's axis. Figure 4.1 illustrates this notation.

During a gear mesh, two tooth flanks initiate contact in form of a contact line (or rather a contact point in two-dimensional space). Let us now consider two involute spur gears rotating in a three-dimensional space, as this is the set-up we will focus on in the future visual analysis.

Figure 4.2 demonstrates the process of gear meshing. For illustrating purposes, we depict helical gears in this figure, because they facilitate the viewing of contact line development. Nevertheless, the principle displayed in the figure applies to spur gears in the same way. The flanks of a pinion tooth and its mating gear engage (Figure 4.2, left), build up a contact line and as the wheels continue to turn the contact line moves

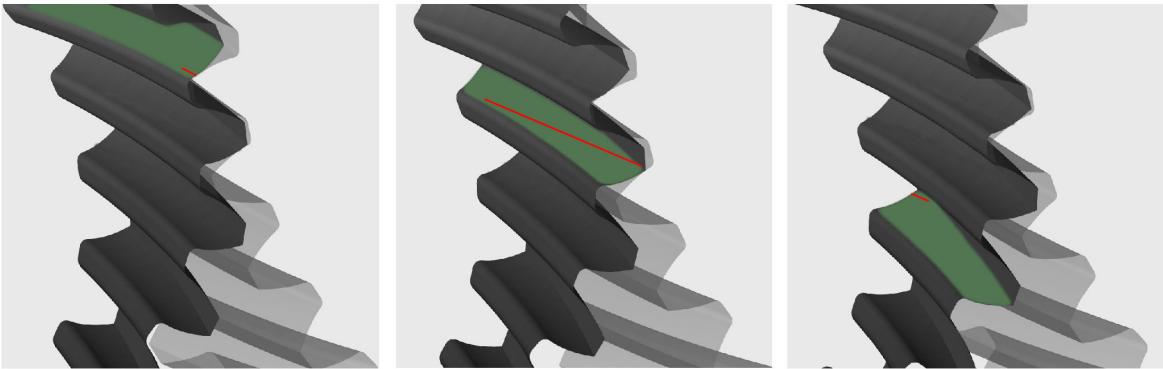


Figure 4.2: Parts of the meshing cycle. Teeth building up contact (left), full contact (middle) and shortly before they diverge (right). The contact line moves along the flank.

along the tooth flanks (Figure 4.2, middle) until the teeth disengage (Figure 4.2, right). This rotation, starting at the point when a single pinion tooth comes across its mating tooth and ending when the teeth separate, is called *meshing cycle* [Flo00]. Before the mentioned flank pair disengages, a second flank pair makes contact. As with the flank pair before, the contact line moves along the tooth flanks during rotation. Two flank pairs are currently carrying the load, which was applied to the gear mechanism. As the wheels turn further, the first flank pair loses contact and the second carries the load alone. Soon, a third pinion flank gets in contact with its mating flank taking over a part of the load. Now two flank pairs are in contact again.

As one can see from the example, the number of flank pairs that are in contact at the same time during the rotation varies. The minimum number of flank pairs that are in contact simultaneously as well as the maximum number that is reached during rotation depend on the geometry of the gear mechanism. The mean value of these two aggregates is called *contact ratio*. For the example used above it has the value 1.5 as sometimes one and sometimes two flank pairs are in contact during rotation.

As mentioned in the beginning of this chapter, we intend to reduce stresses, which may lead to fatigue or failure. An even distribution of contact forces, along a tooth face as well as along flank pairs, is of great use in this case. Thus, the more flank pairs are in contact at the same time, the better the forces are distributed. This is roughly modeled by the contact ratio: the higher the value, the more flank pairs are in contact, meaning a distribution of contact forces to many flank pairs. We can also state that longer contact lines of flank pairs result in a smoother motion transmission, as contact forces are better distributed along the tooth flanks.

Keeping these characteristics in mind, we take a look at the simulation of cylindrical gears described in the following section. This section also presents how surface ensembles are constructed from the resulting simulation data.

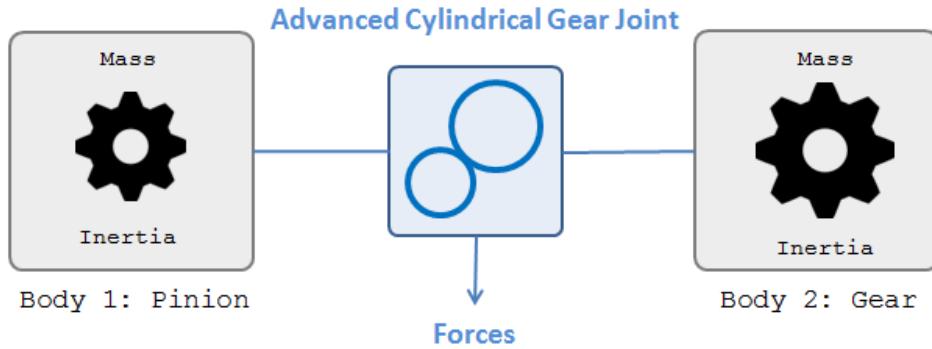


Figure 4.3: Modeling a gearing system: bodies carrying physical properties (left, right) are connected by a joint that produces forces (middle).

4.2 Simulation and Data Conversion

Gear operation depends on different parameters, whose configuration defines in which way the gear is transmitting motion, for example if a change in direction or speed is obtained. We aim to analyze the stresses on the wheels' teeth, which arise when a gear system is put into operation. Therefore, simulation is the instrument of choice to investigate different initial gear system states and the forces that arise on the gears.

We use the AVL EXCITE Power Unit solver [Gmb] to perform the simulation of our gearing system. It is commonly used in the context of engine design, but also supports analysis and optimization of existing engines, transmissions, and powertrains [Gmb12]. Multi-body dynamics environments like EXCITE model gearing systems by using *bodies*, which represent the mating gear wheels, as well as a *joint*, which serves as a connection of two bodies (Figure 4.3). Bodies carry the physical properties of the gears, such as mass and inertia, while the joint results in forces as the bodies are set in motion (M. Sopouch, personal communication, December 4, 2015).

However, simulation cannot start until we define its input. For simulation of the gearing system, we specify gear geometry as well as varying run-time parameters and design parameters for multiple simulation runs. The simulation model used in this case study does not correspond to realistic cases, which contain much more complex compositions of gears. Instead, we use a more idealistic model, cut down on the basic principles of gearing and with little complications caused by misalignment and others.

Gear Geometry, Input Parameters and Simulation Results

The gearing system consists of two involute spur gears, which mesh in the form of outer gearing. The pinion has 21 teeth, while the gear holds 31 teeth, corresponding to tip diameters of 13,8 cm and 19,8 cm. All examinations are carried out using an internal joint coordinate system, which is depicted in Figure 4.4. Its origin corresponds to the pinion's center. The x -axis is build up by the vector pointing from the origin to the gear's center, while the z -axis agrees to the pinion's axis of rotation. The y -axis results from the other axes by assuming a right handed coordinate system [Gmb12].

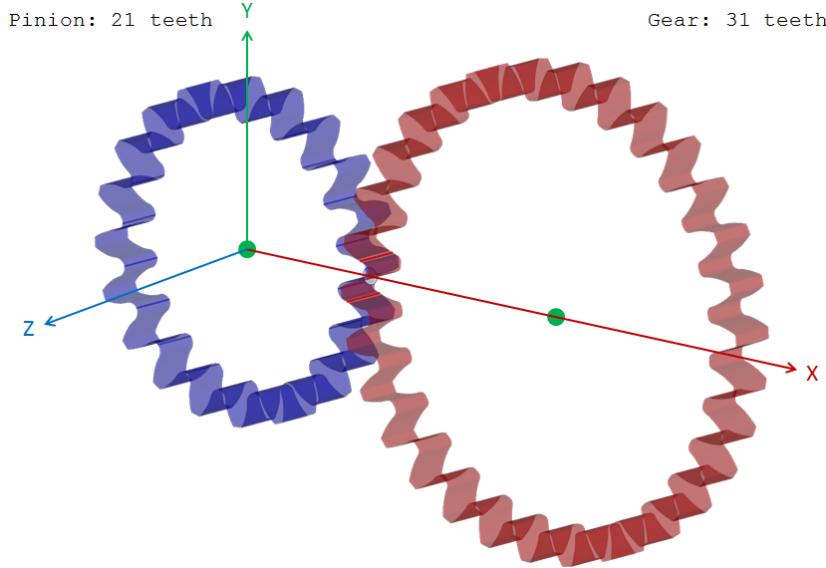


Figure 4.4: The gearing system: two involute spur gears with 21 and 31 teeth are mounted on parallel axes. Positions are given with respect to the joint coordinate system.

Contact lines are discretized by regular sampling along gear width, which is a consequence of discretization of gears into a series of slices [SBP⁺12]. This results in every contact line being represented by 21 contact points with equal x - and y -coordinates and varying z -coordinate. This number of discrete points per contact line is in no way related to the number of the gears' teeth.

Various control parameters can be defined for simulation of a gearing system. The solver takes nine input parameters, which describe the properties of gear operation.

Two run-time parameters describe operating conditions by means of *angular velocity*, which specifies the rotational speed of the gears, and *load torque*, which is applied to the gear shafts. Both are varied three times each, resulting in nine possible operating conditions. For exact values used for speed-load-combinations see Figure 4.6, left.

When designing gears, tooth profiles are modified in different ways to compensate irregularities and to avoid high local edge and tip loading. At the same time, gear production includes manufacturing tolerances, which allow a certain number of gear profile errors. Visual analysis later helps to identify critical profile errors leading to undesired forces, which supports the determination of tolerances that have to be reduced for further production. Seven design parameters model such profile corrections and profile errors:

- *Tip Relief Pinion* ($z1_C_\alpha_a$) (Figure 4.5a),
- *Pressure Angle Correction Pinion* ($z1_C_\alpha_h$) (Figure 4.5b),
- *Barrelling Pinion* ($z1_C_\alpha$) (Figure 4.5c),
- *Crowning Pinion* ($z1_C_\beta$) (Figure 4.5d),

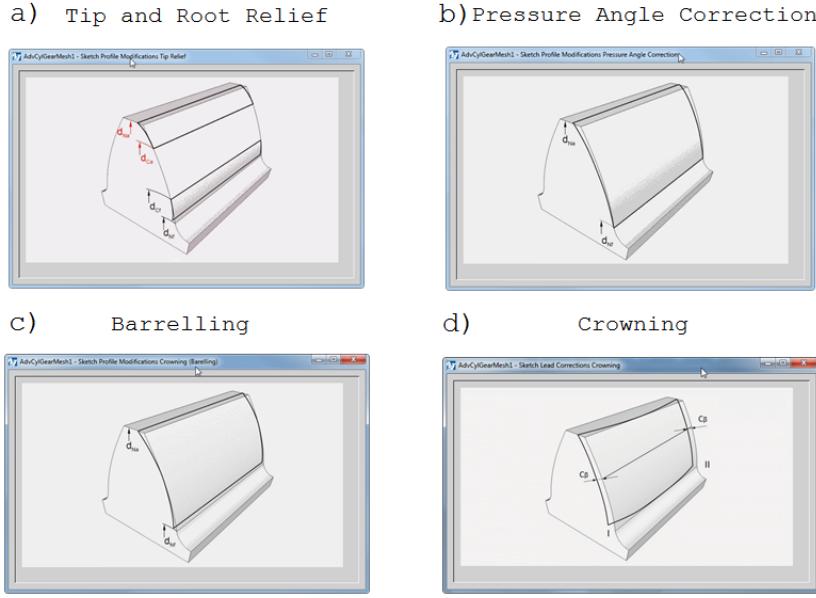


Figure 4.5: Standard profile modifications in lead and profile direction, used for compensating irregularities in gear performance, are combined to 100 different gear designs.

- Pressure Angle Correction Gear ($z2_C_Halpha$) (Figure 4.5b),
- Barrelling Gear ($z2_C_alpha$) (Figure 4.5c),
- and Crowning Gear ($z2_C_beta$) (Figure 4.5d).

We use Sobol sequences to generate 100 combinations of the seven-dimensional design parameter space per operating condition. Table A.1 depicts the parameters' exact values. This yields $3 * 3 * 100 = 900$ possible parameter combinations in total. The control parameters are also stored in the final data set as scalar attributes. For each of the parameter combinations, we execute one simulation run that covers a simulation period of one pinion revolution. The complete simulation of 900 runs takes about one hour on a typical PC.

We compute different response parameters for each of the 900 simulation runs. 1173 time steps are simulated. Each of them corresponds to a rotation angle of the pinion, to which we refer as reference angle. Simulation starts with a reference angle of 1080 degrees, which is increased by 0.261 to 0.3072 degrees every time step, resulting in an angle of 1440 degrees at the last simulation step.

For each step, the simulation computes the positions of the current contact points, as well as forces that arise at each point, such as contact pressure, friction force, normal force, and others. We also get the values of other gearing properties, for example mesh deformation or total forces, which are added up along right and left flanks.

Using the given geometry and a defined visualization angle range, we use an automatic data processing to convert this simulation output to attributes of several data types,

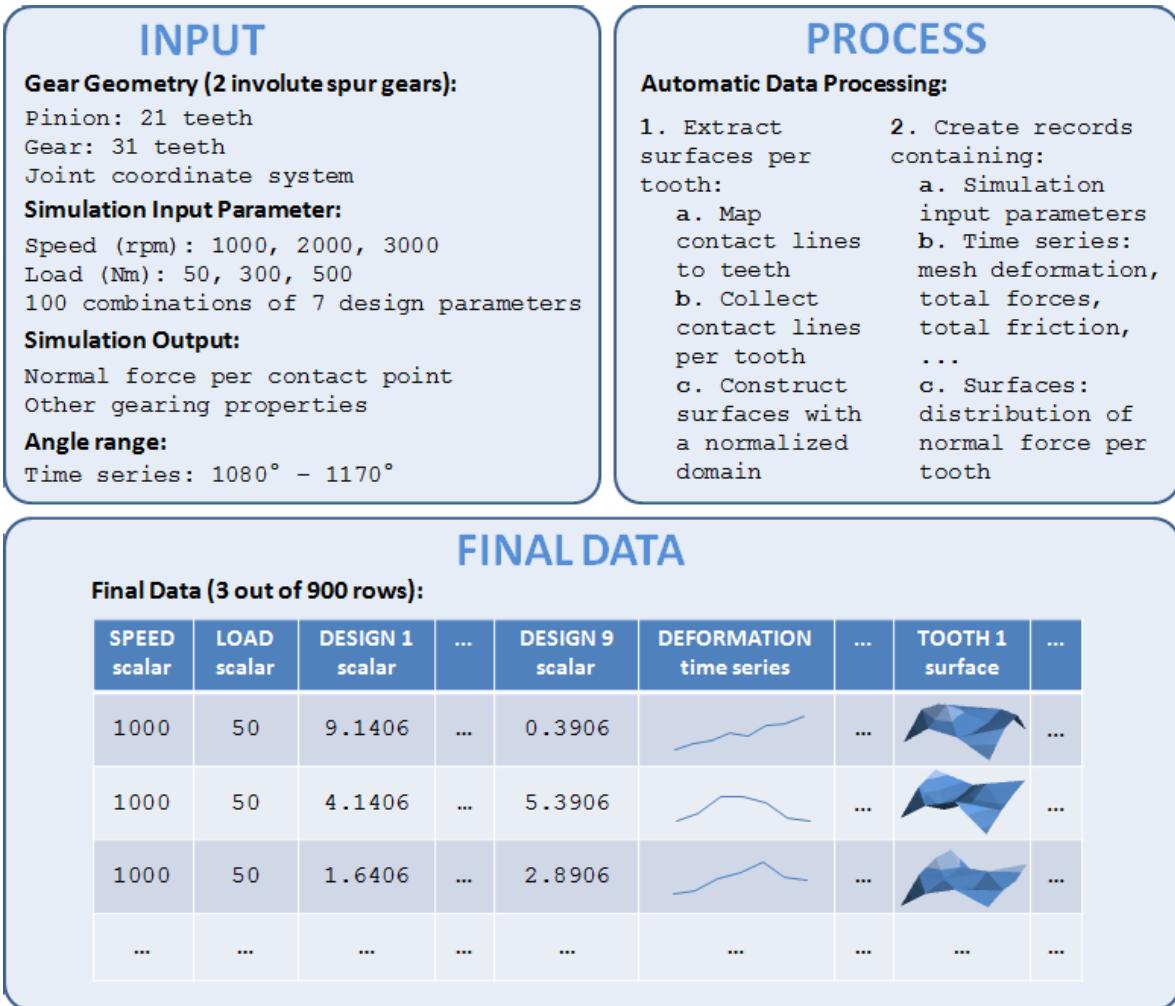


Figure 4.6: Gear geometry, simulation parameters and defined visualization angle range (top left) are automatically processed (top right) to generate the data set used for evaluation. The data table illustrates how final data are organized (bottom).

which are added to the scalar control parameters to form the final data set (Figure 4.6, top). An extract of the final set of data points is depicted in Figure 4.6, bottom. For simplification purposes, we solely consider response parameters measured at left pinion flanks for data conversion. Scalar attributes are obtained by computing global statistical values from gearing properties measured at each contact point. One example is the maximum contact pressure, taking every contact point at every time step into account. We also integrate complex data types into our data set.

Therefore we compute time series data from quality measures, which are simulated for all left and right flanks respectively. This is realized using reference angle as x -dimension and the measurement as y -dimension. Only simulation results with a reference angle between 1080° and 1170° are taken into account. In this way, we obtain pairs of reference angle and mesh deformation for right flanks, for example.

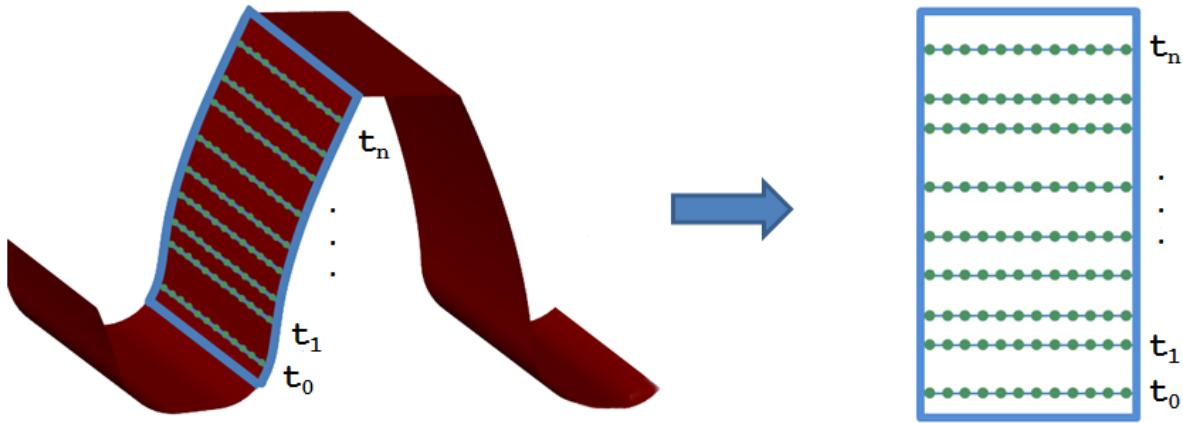


Figure 4.7: Surfaces are constructed by collecting contact lines, that move along the tooth flank over time (left). Each contact line is discretized with 21 equidistant sample points over gear width, which form a rectilinear surface grid (right).

Based on the positions of the contact points and the measured forces for each point, we implement multiple conversion steps to construct surfaces per pinion tooth, representing stresses that arise on the specific tooth during meshing (Figure 4.6, top right). The following paragraph describes the conversion to data surfaces in more detail.

Surface Ensembles

As outlined in Chapter 1, we intend to evaluate our interactive visual analysis approach focusing on surface ensembles. In addition to the scalar and time series attributes described in the previous paragraph, we extract data surfaces from simulation data. In Section 4.1, we mention that the contact line of two engaging teeth moves along the pinion tooth flank as the wheels turn (see Figure 4.2). For surface data construction per tooth, we consider all contact lines arising on an individual tooth over time. Remember, that a contact line is discretized with 21 equidistant contact points over the gear width. The lines are then positioned on top of each other, with respect to their specific position on the tooth flank, resulting in the contact points making up a rectilinear grid as shown in Figure 4.7. Measured force values for each of the grid points form the data surface.

First of all, we determine frame conditions of the data conversion:

- (1) Of all meshing properties that are computed during simulation, we choose normal force for visualization. It has the most significance when investigating gear stresses, such as flank contact pressure as well as root bending stress (M. Sopouch, personal communication, December 4, 2015).
- (2) We choose a range of visualization angle, which refers to the pinion rotation and determines which parts of the simulation results are considered for data processing. This means that only contact lines and forces corresponding to a reference angle that lies within this range are considered. We define one revolution of pinion for visualization, which corresponds to a reference angle range of 1080° to 1170° .
- (3) To be able to construct data surfaces per tooth, for every contact line we need

to know on which specific tooth it arises. Therefore we establish a mapping between contact lines and their corresponding pinion teeth as prerequisite for surface extraction.

The basic idea for the mapping is to find a representation for each pinion tooth, compute distances from contact lines to teeth, and take the line-tooth assignment that results in the lowest distance. First, we considered to model a single tooth by using a cutting plane that contains the pinion's center and divides the tooth into right and left halves. This would require following computations to be performed in three-dimensional space. To simplify the mapping, we transfer the gearing composition from 3D space to 2D space and represent teeth using lines instead of planes. To achieve an undistorted transformation, we project the gear wheels and the contact lines on to the plane of which the contact lines are normals. This results in a 2D point representing initiated contact as mentioned in Section 4.1.

As a rough model of the pinion's teeth in 2D space, we assume lines from the pinion's center to each of the pinion's teeth, resulting in a star with 21 rays (Figure 4.8). Having one line representing the first tooth, we can compute representative rays for all other teeth by rotating the first ray. The rotation angle is determined by the number of pinion teeth, resulting in $360/21 = 20$ degrees. However, we are missing a starting point for ray construction by rotation. This issue is solved by taking the 2D point representing the first arising contact and assigning it to tooth one. The point also determines the ray that represents the first tooth. This is an appropriate procedure, because we only need a rough assignment of contact lines to teeth and no exact positions or measurements. It is therefore not mandatory that the rays pass through the tip of each tooth.

For further simplification of the mapping, we do not consider rays in the form of a position vector and a directional vector, which would require the computation of point-to-plane distances. Instead, we take the endpoint of each ray as reference and use the Euclidean distance between a contact point and the endpoint of a ray for distance computation. This means that we do not store complete rays, but only the starting contact point and the results of its rotation.

For every time step we now read in the contact lines that are currently present. For each of their representative 2D points we compute the Euclidean distance to each of the tooth rays. Then we assign the contact line to the tooth that is modeled by the ray that has the lowest distance to the contact line's representative 2D point. After having processed a time step the pinion model has to be rotated around the same angle as the pinion in simulation is rotated.

In this way, for each contact line that is generated during one complete revolution, we obtain an ID ranging from 1 to 21 that indicates the tooth, on which the particular contact line arose. Knowing the corresponding pinion tooth for each contact line, we can use simulation results to extract data surfaces, which depict the distribution of normal force over the entire tooth flank.

For each tooth, we collect all contact lines that were assigned to it. For comparison of multiple surfaces, we have to assure that identical (x, y) -coordinates of the surface grid represent the same locations on different tooth flanks. Thus, we have to define all surfaces in an identical domain. As all teeth have the same dimensions, we use tooth

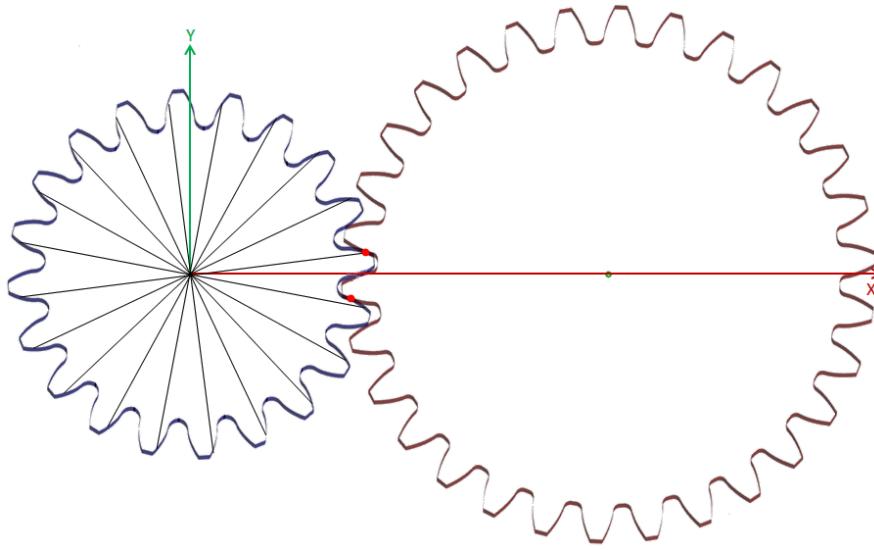


Figure 4.8: The gearing system projected on to xy -plane. The red points represent contact lines. Rays model the pinion’s teeth for mapping of contact lines to teeth.

width and height as x - and y -axes for surface construction. Grid points along tooth width are given by discretization of contact lines into 21 equidistant contact points (see Section 4.2). In this way, they make up the x -axis of the surface coordinate system. The y -coordinates of grid points are obtained from the contact lines’ relative positions on the tooth flank. For the following explanations we refer to the 2D space that was already used for the mapping of contact lines and teeth. As a reminder, a contact line is represented by a 2D point in this case. We use a straight line in the direction of pinion center to tooth tip to approximate the originally curved tooth flanks, as shown in Figure 4.9. The tooth height h along this line can be computed by subtracting the root radius of the pinion from the tip radius resulting in $h = r_{root} - r_{tip} = 6,9\text{ cm} - 5,923\text{ cm} = 0,977\text{ cm}$. For computation of the relative value of an individual contact point P , we first compute its distance d to the pinion’s center M . By subtracting the root radius from the resulting distance, we obtain a value y between 0 cm and 0,977 cm and thus map the contact point onto the line approximating the tooth flank. The projected contact point is represented by P' in Figure 4.9, right. The value of y represents the relative position of the contact line along tooth height, which determines at the same time the y -coordinate of the grid point.

By adding the corresponding force value on the z -axis for all grid points we build up a data surface. Having constructed one such data surface for each tooth, we obtain a surface ensemble by observing a complete gear-wheel.

Finally, in the following section, we evaluate the methodology (Chapter 3) by performing a visual analysis using the surface ensembles obtained from the simulation data.

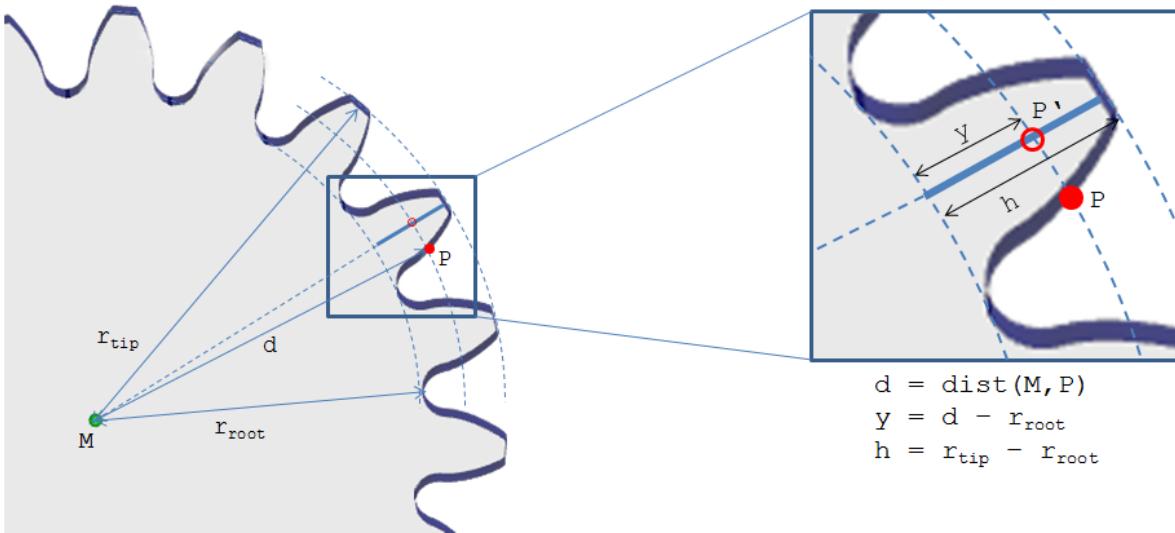


Figure 4.9: Transforming contact lines to a normalized domain, which is equal for all teeth. Contact points are mapped to a straight line, that approximates the curved tooth flank, to obtain a normalized y -value.

4.3 Interactive Visual Analysis of Gear Stress

As gears have become an integral part of mechanical engineering, their reliability is of central importance for a proper functionality of various machines. The first step when designing gears is the definition of basic gear dimensions according to requirements, such as the desired transmission ratio and maximum allowed space for packaging, that emerge from the target application. Gear contact analysis, which is performed in the next step, supports domain experts in:

- Achieving sufficient safety factors for strength and durability,
- Achieving noise and vibration targets,
- Minimization of gear failures,
- Improving gear efficiency,
- Increase of gear service life.

Traditionally, gear design is carried out in iterative cycles of simulation, gear contact analysis, and adjustment of gear geometry in terms of profile modification according to analysis results. This approach does not support analysts in exploring results from multiple simulation runs with different geometry settings. Based on multiple simulation runs, the interactive visual analysis methodology presented in this thesis allows for simultaneous exploration and investigation of various gear geometry settings. In this way, new insights into the impact and significance of changes in gear geometry on resulting forces, which arise across tooth flanks, emerge.

Let us take a look back at Chapter 1, where we introduced two main questions, which form the basis for analyzing simulation results and corresponding inputs. The first one addresses the sensitivity of simulation output towards varying input parameters. Secondly, analysts aim to find configurations of input parameters that produce desired simulation results. Such tasks also apply to the field of gearing. In collaboration with domain experts, we developed specific analysis tasks regarding gear contact analysis:

- **T1: Multi-resolution overview** – familiarize with the given data set. Get a first notion of possible analysis workflows.
- **T2: Sensitivity analysis** – how do profile modifications and errors affect the distribution of forces?
- **T3: Characterization** – identify characteristics of force distribution, e.g. detect peaks indicating extremely high forces.
- **T4: Regions of interest** – identify designs leading to regularly distributed forces. Avoid control parameter settings leading to high or fluctuating forces. Minimize gear failure, noise, transmission error, and others.
- **T5: Side-by-side comparison** – compare one gear design to another one.
- **T6: Flexibility** – allow constant changes of perspective, e.g. between investigation of all teeth per simulation run and all simulation runs for a selected tooth.

Before dealing with particular analysis tasks and performing a target-oriented exploration of the data set, visual analysis starts by gaining a fundamental overview of the investigated gearing system (T1). To obtain a basic understanding of how the meshing cycle behaves and to ensure that the implemented methods give correct results, we analyze an idealized gear model as reference case. For this case, only basic gear geometry, including type of gearing (spur or helical), number of teeth, helix angle, gear width, and tip diameter, is considered. Profile modifications and errors, represented by design parameters as described in Section 4.2, do not play a role in this initial case.

Figure 4.10 shows the reference case in terms of normal mesh deformation and normal force, which are stressing the teeth. Normal mesh deformation is added up across the left flanks of all pinion teeth and displayed as a function over time in curve view (Figure 4.10, top). It forms one part of the overall transmission error, which represents the difference between the actual position of the pinion and the supposed position under ideal conditions. The transmission error is the key excitation mechanism for gear noise and vibration and cannot be avoided due to the nature of gear design. Normal mesh deformation arises due to variation of the number of flank pairs that are in contact during revolution, causing a periodic fluctuation of contact forces – and consequently deformation of teeth (Figure 4.10, top).

For closer investigation of a single pinion flank, we use the 3D surface display in combination with profile curves and contours, depicted in Figure 4.10, bottom. All of them

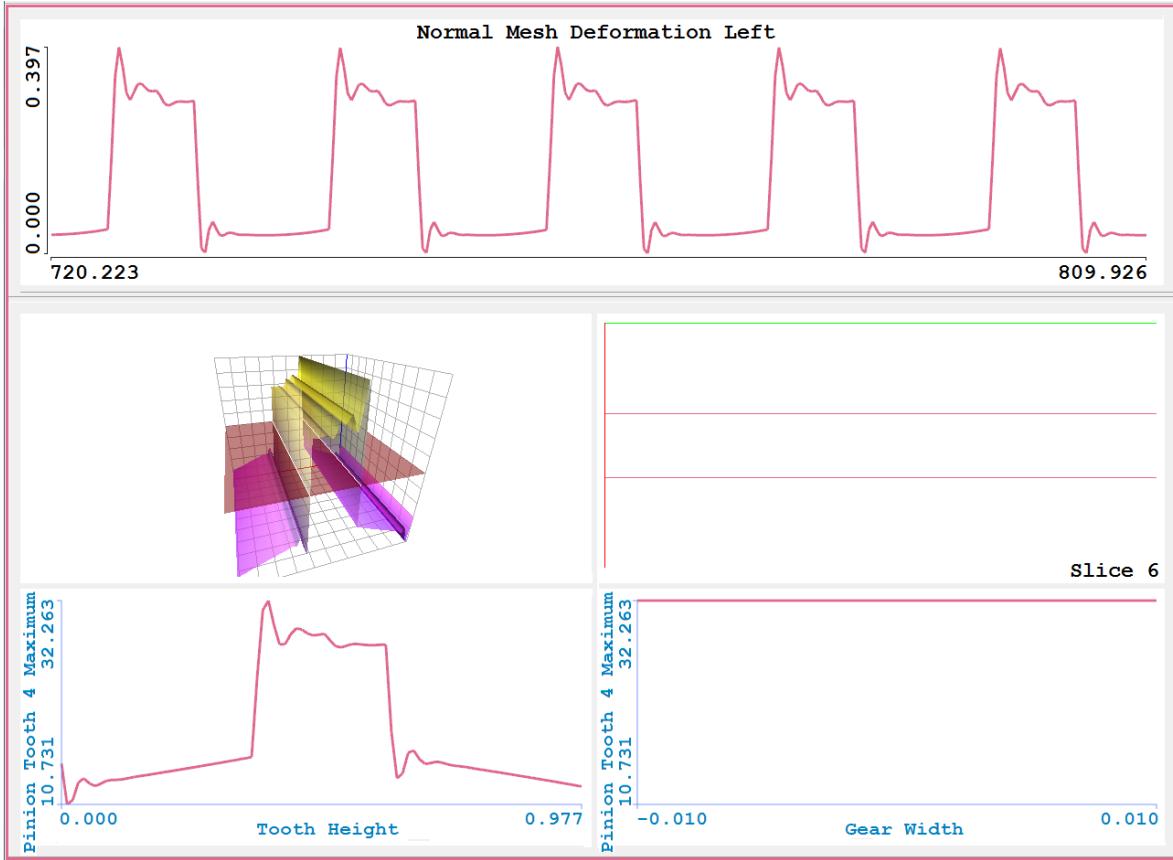


Figure 4.10: Reference case: curve view depicts periodic fluctuations of normal mesh deformation (top). Profile curves show increased normal force caused by contact ratio (bottom left). It is constant over gear width due to ideal conditions (bottom right).

show normal forces, which are stressing the flank of a selected pinion tooth.

The lower left curve displays an aggregated distribution of normal forces along tooth height, by summarizing normal force values along lead direction using maximum (Figure 4.10, bottom left). We can identify increased normal force in the central section of the flank surface. This phenomenon can be explained by the fact that, depending on one or two flank pairs being in contact, normal forces are higher or lower due to distribution of load to one or multiple flank pairs. They will be the highest when only one flank-pair is in contact, which then carries the load alone. Such sharp transitions of normal force due to contact ratio are not desired in gear design, which is why geometry optimization in profile direction should be utilized to avoid them. We can also spot smaller fluctuations after normal force rapidly increased or decreased, which quickly steady (Figure 4.10, bottom left). They mark positions, where the number of flank pairs in contact suddenly changes. If teeth were considered fully rigid, this curve representing normal force would be step-wise and the characteristics of varying force in profile direction would be rather monotonic. Since the simulation takes a mathematical model for elastic structures into account, tooth deformation depends on the distance

from the root and varies also depending on tooth thickness. Therefore, also higher order components of normal force, caused by engagement and relief of the second flank pair in contact, become evident.

As mentioned above, this reference case originates from gears being simulated without profile modifications or errors under ideal operating conditions. As a consequence, normal force is constant over the gear width in lead direction, which can be seen at the straight horizontal line in the corresponding surface profile (Figure 4.10, bottom right). However, realistic gear system are flexible and include misalignment of teeth, which results in high edge loading if the tooth profile will remain constant in lead direction. To avoid gear damage and failure that can be caused in this way, tooth profile optimization should also include modifications in lead direction.

While gaining a comprehensive overview of the gear system's behavior when working under ideal conditions, we identified several problematic issues causing reduced gear performance with respect to durability, noise, efficiency, and failure, for example. To overcome these problems, we apply appropriate tooth profile modifications in profile and lead direction with the aim of improving gear mesh behavior under different operating conditions. To evaluate different gear designs, we use the interactive visual analysis approach presented in this thesis in order to solve analysis tasks as listed above.

As a first step towards sensitivity analysis (T2), we intend to explore the distributions of normal force. Three-dimensional box plots form a suitable starting point for analysis, as they depict the distribution of normal force across the flank of a selected tooth at a high level of abstraction (T3). Statistical measures enable a rough assessment of arising forces at different sections of the tooth flank. Extreme values, in particular maximum forces, can be quickly recognized in this view, which allows for an efficient detection of critical parts of the flank that have to withstand high stresses as well as high edge loading, for example. When used together with brushing, this view enables an efficient identification of parameter settings that lead to regularly distributed forces (T4) at a chosen level, which is determined by the selected number of subdivisions.

Figure 4.11 shows two brushes, displayed with four times four three-dimensional box plots each. The brush on the left results in regularly distributed, low forces. When considering the context box plots, we can see the amount to which brushing has evened out the differences in forces. Even at a more detailed level, which is represented by additional two-dimensional box plots (Figure 4.11, bottom left), forces are regularly distributed in the lower area of the scale. In contrast to that, brushing low barrelling values results in a distribution of forces that still contains high local differences and edge loading (Figure 4.11, right).

Heat maps allow for exploration of normal force distributions from another point of view. Where box plots require the selection of a single tooth that is to be investigated, we can make use of heat maps that are based on a run-centric approach as we described in Section 3.1. They display the distribution of forces over all pinion teeth for a small number of chosen gear designs. Figure 4.12 depicts the effect of profile modifications in the form of contact forces, which arise at the flank of each pinion tooth, under three different loading conditions.

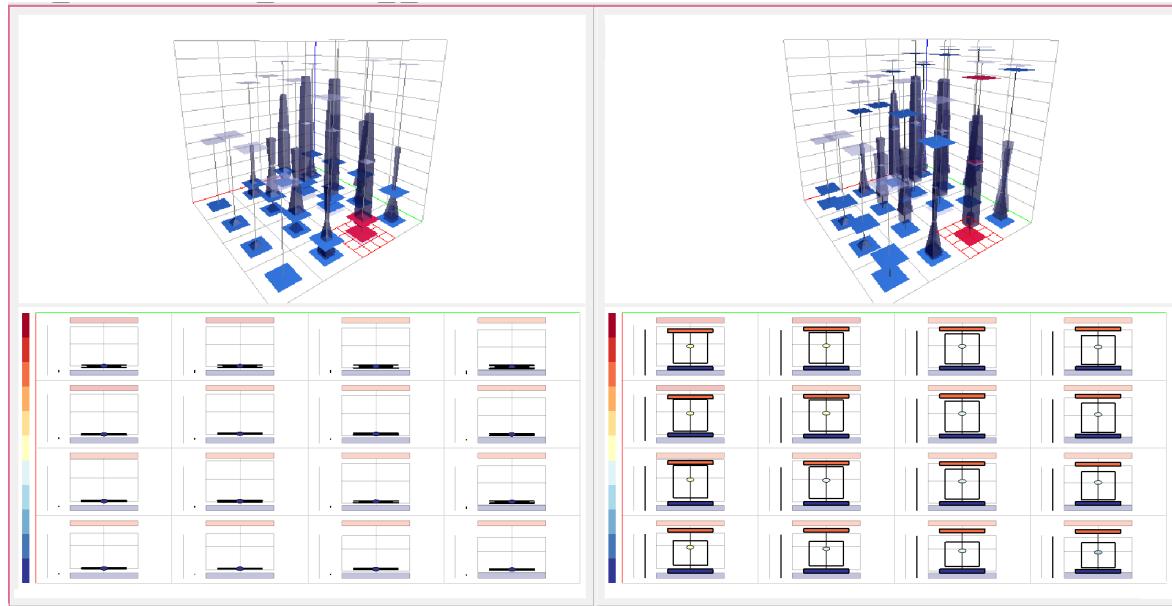


Figure 4.11: Box plots support identification of (un)desired distributions. Low and regularly distributed forces can be selected (left), in the same way brushing low barrelling values results in local differences and edge loading (right).

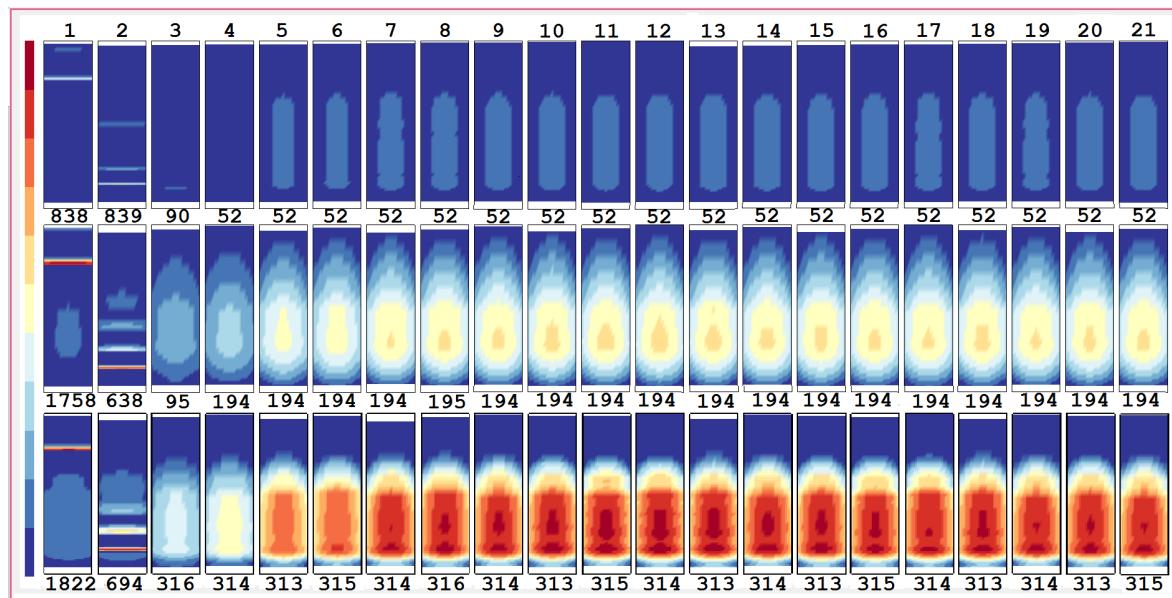


Figure 4.12: Run-centric heat maps illustrate the effect of three different loading conditions. High load causes extremely high contact forces (bottom). Sharp impacts at first two teeth appear due to closing backlash caused by application of load (left).

As we can easily see, contact forces depend on the amount of torque transmitted by the engaged gear pair. Colors indicate extremely high forces for a load torque of 500 newton metre (Figure 4.12, bottom), while lower load torques of 50 and 300 newton metre result in acceptable forces that do not stress the tooth flanks in a critical way (Figure 4.12, top and middle). The simulation starts in a position where tooth flanks of pinion and gear are not in contact. By applying load torque on the pinion, the backlash (i.e. the clearance between mated gear teeth) between driving and driven flanks is closed, which results in sharp impacts that appear at the first two pinion teeth. It can be seen as horizontal lines in the first two heat maps of each row in Figure 4.12. After this initial simulation phase is finished, the pattern of gear contacts becomes steady. The latter can be in particular observed in heat maps for medium and high load torque, where forces form regular ellipses across the remaining tooth flanks. Due to crowning, which was applied as profile correction in lead direction, the distribution of forces along gear width is not uniform anymore and shows a preferable reduction of edge loading, which leads to such shapes. At the same time, barrelling, which was applied in profile direction, reduced the sharp, step-wise transitions of normal force due to gear contact ratio, which could be observed in the previous reference case.

After having explored the overall distribution of forces and impacts of fundamental settings, we aim to identify critical combinations of design parameters that lead to undesired scenarios (T4). This allows us, among others, to determine the range of allowed profile modifications in gear design as well as tolerances that can be included in the manufacturing process without causing intolerable gear mesh behavior.

All varied profile modifications are shown in the form of parallel coordinates in Figure 4.13, left. Normal force, added up across all tooth flanks of pinion, is depicted as time series in a curve view (Figure 4.13, right). Again, we can clearly observe three clusters of curves originating from three different loading conditions that were simulated. Among the curves representing high load torque, we can recognize several cases leading to an extreme increase of contact forces at particular time steps. By brushing those peaks in curve view we can identify critical combinations of design parameter values that result in extreme tooth loading. Looking back to the parallel coordinates view where brushed parameters are highlighted, we can detect patterns among the profile modifications of the two brushed critical cases. For both cases, we applied low values for both pinion and gear barrelling as well as high values for gear crowning. Grey lines show all remaining combinations of profile modifications, offering the possibility to relate the selected data points to the remaining cases of the data set. Here, we can carefully conclude that low barrelling and high gear crowning values are the major causes for high forces, while other control parameters seem to have a minor impact. This also gives an indication for requested minimum values for barrelling as well as upper limits for gear crowning values to be used in future gear designs. These critical cases, which we identified using standard views, are set as the starting point for further investigation.

In the next step we use the three-dimensional surface display combined with surface profiles and contours, depicted in Figure 4.14 on the left, as well as heat maps, shown on the right, for detailed analysis of the selected critical operating conditions. These

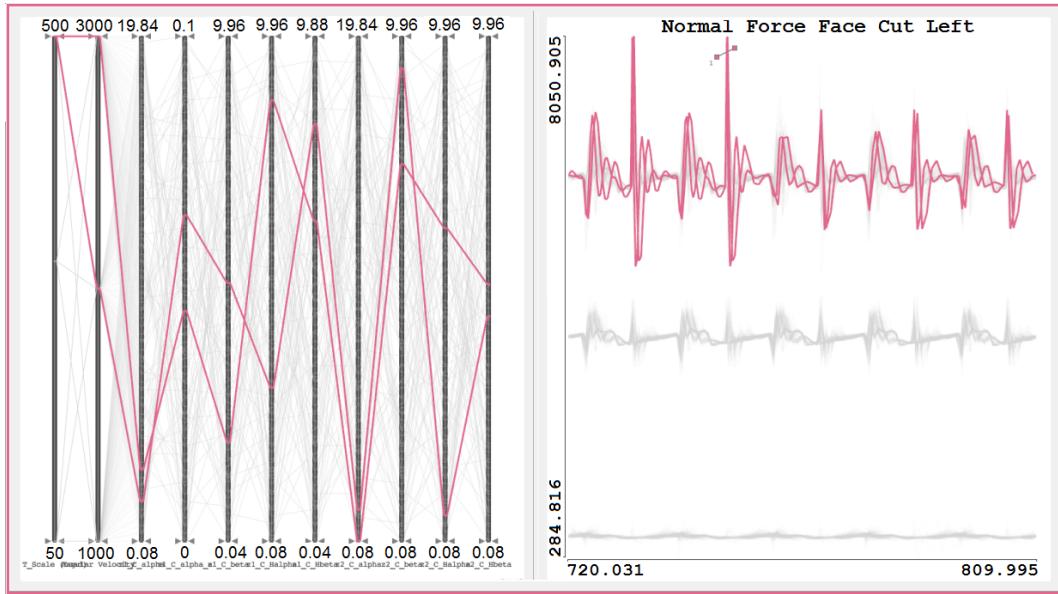


Figure 4.13: Parallel coordinates showing 900 investigated gear design variants (left). Brushing extreme increases of contact forces in curve view (right) results in highlighted design parameter combinations of two critical cases (left).

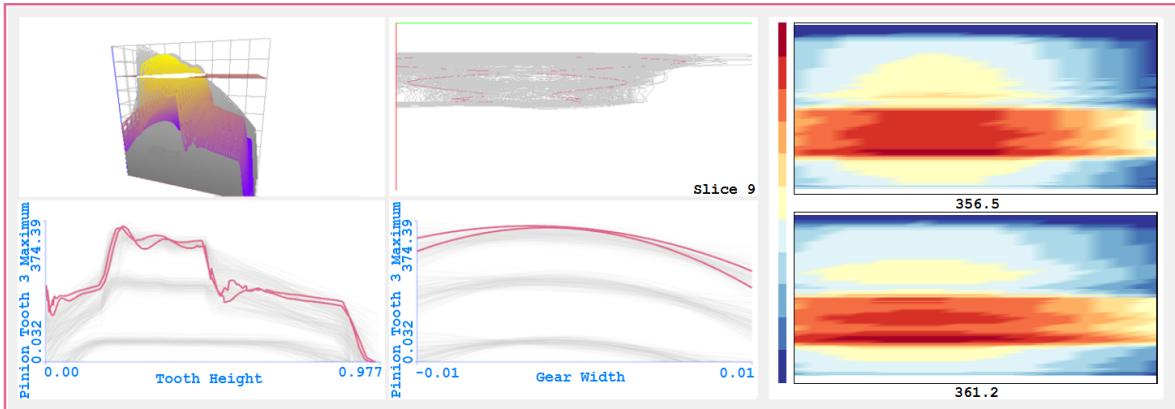


Figure 4.14: Detailed analysis of third tooth for critical cases: all displays reveal high forces shifted to the left, called egde loading. Remaining teeth show similar behavior.

views allow for investigation of a specific pinion tooth that the analyst is interested in. When selecting the third pinion tooth, all displays show high forces shifted to the left of the flank for brushed data points, revealing higher edge loading at one side of the gear. This is a consequence of misalignment of gear teeth, which can typically be solved by additional gear flank profile corrections in different sections of the gear. When scrolling through the remaining pinion teeth, we can observe similar behavior.

To verify our hypothesis that such high forces and edge loading are caused by the design parameter values that we identified, we consider heat map view to compare selected cases with a subset of converse operating conditions (T5).

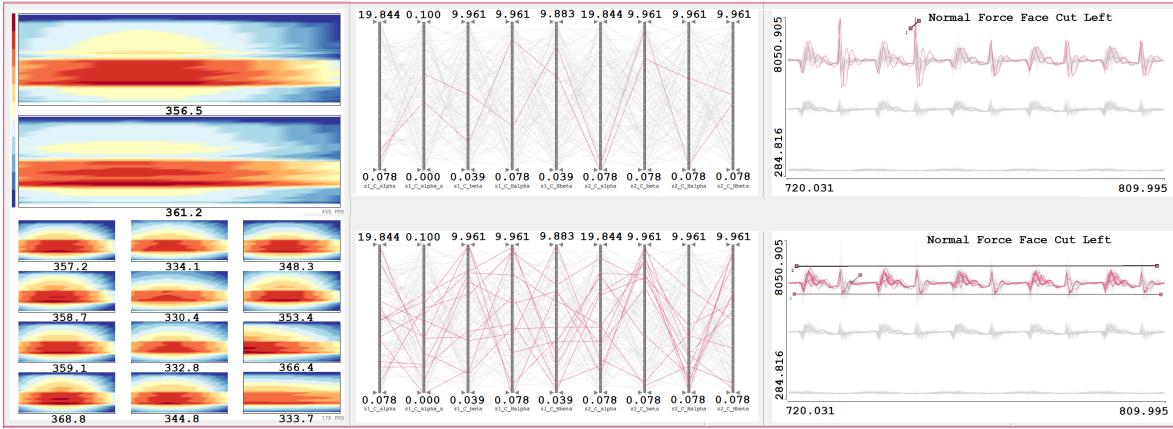


Figure 4.15: Direct comparison of critical cases (top left) to a set of reference designs (bottom left) using heat maps. Parallel coordinates (middle) and curve view (right) show corresponding brushes.

Figure 4.15, top, shows the selected critical cases, while the opposed subset of data points is displayed at the bottom. For both subsets, parallel coordinates (Figure 4.15, middle) depict the corresponding values for design parameter, while the corresponding brush is seen in curve view (Figure 4.15, right). We can see that edge loading is not limited to the two suspicious design parameter combinations. The distribution of forces is similar for both sets of surfaces. Nevertheless, we can observe that the sharp transitions of normal force along tooth height, which arise for critical cases (Figure 4.15, top left), are reduced in the subset of converse operating conditions (Figure 4.15, bottom left). This is due to the higher barrelling values that were applied for the latter. As can be gathered from the displayed maximum value per surface, the critical cases do not have a significantly higher maximum force, although some of the converse conditions present maxima of a slightly lower value.

This comparison is limited to a selected tooth of the pinion, which means that all heat maps represent the same tooth, but for different brushed simulation runs. We also want to take a look at all teeth for selected simulation runs simultaneously, to visually compare overall impacts of different gear designs (T6). For this purpose, we can switch heat map view to run-centric mode, resulting in all teeth per brushed simulation run being displayed. We select one of the critical cases to be compared to one design of the previously used subset of converse gear designs (Figure 4.16).

We can observe a regular pattern of contact stress across every pinion tooth for both simulation runs. In general, the critical case (Figure 4.16, top) demonstrates higher forces arising at the tooth flanks than it is the case with the opposed gear design (Figure 4.16, bottom). Furthermore, we can identify much sharper transitions of forces in profile direction for the critical combination of profile modifications, just as we have already seen in Figure 4.15, which can again be explained by low values for both pinion and gear barrelling.

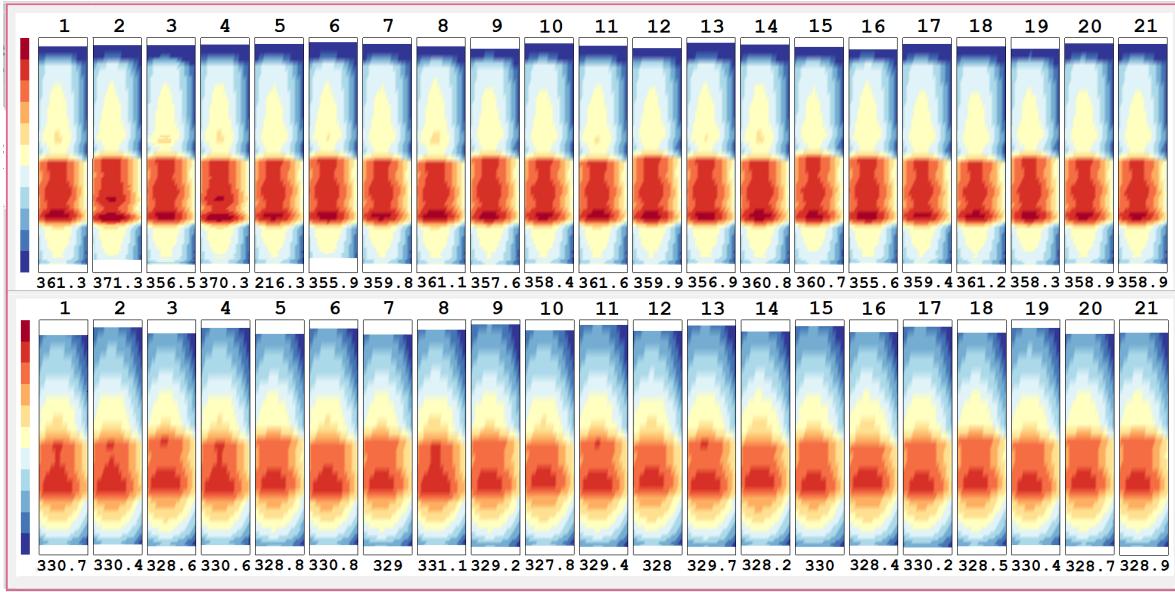


Figure 4.16: Side-by-side comparison of a critical case (top) to another design (bottom). The critical case reveals higher forces and sharper transitions along tooth height.

The visual analysis approach presented in this thesis allows for efficient, simultaneous exploration and analysis of various gear designs, including basic geometry settings as well as advanced tooth profile modifications, in an interactive way. It holds the potential to support the detection of meaningful correlations and patterns by using different perspectives that can be considered for data analysis. Domain experts stated that the presented methodology adds value to existing traditional methods in terms of interactive investigation of a large number of different design variants. Next steps would be to use not only normal force as target variable, but to also add other important physical quantities, such as contact pressure, friction force, sliding velocity, and others, represented by surface ensembles, to the data set. In this context, one could also allow for combination of these quantities by mathematical operations, supporting the investigation of different failure modes.

5. Conclusion

The analysis and exploration of data originating from advanced multiple-run simulations is a challenging task. In particular, simulation of data surfaces leads to increased data complexity, which stresses the analyst’s cognition and perception. We introduce an interactive visual analysis approach that integrates novel aggregation techniques and corresponding visualization for analysis and exploration of surface ensemble data. Data can be considered in two different ways: (1) focusing on individual simulation runs or (2) with regard to specific surface attributes. Conventional methods often consider each dimension in simulation data to be a scalar value and thus break down the original complex simulation results to scalar components. In contrast, our methodology relies on a coherent representation of individual data surfaces to enable a thorough understanding of complex relations and interplay of input parameters and simulation results. At the same time, it also addresses the analyst’s mental model of data items, which reduces the cognitive load.

Descriptive statistics, displayed by three-dimensional multi-resolution box plots, convey a rough notion of the distribution of values across ensemble members at the beginning of analysis. Overall patterns like shadow stripes in light distribution can be detected. Two-dimensional box plots support a task-oriented investigation of individual sub-domains by additionally providing information about further subdivisions. This includes color-coded statistics as well as exact values on mouse-over. However, box plots are not accurate enough to enable investigation of surface ensembles in greater depth for different reasons. Occlusion and clutter emerge, when the number of sub-domains exceeds a certain limit. We also do not have an individual representation for each ensemble member, which limits a detailed analysis of particular surfaces.

Thus, we provide visual representations for each surface in the form of heat maps that are easy to interpret due to color-coded function values. A hierarchical arrangement enhances the side-by-side comparison of surface subsets by enabling the analyst to manually define a context ensemble, which can then be simultaneously observed with a current brush. A third display provides screen space for detailed analysis of a single

surface. The heat map view has been used throughout the analysis process, to gain an overview of the data set on the one hand, but on the other hand it was also considered for detailed investigation of regions of interest.

Contours arising from intersection of a surface ensemble with a horizontal cutting plane were intended to provide for an additional perspective to support the mental assignment of surface parts to a continuous shape. However, it turned out that contours could not add significant value to the perception of surface distribution, as surface profiles alone provided enough information to guess the surfaces' shapes.

Using a three-dimensional surface display, we also present data surfaces in their original form and thus contribute to building up a mental model of individual surfaces' shape. A transfer function that maps surface points that are closer to the camera to higher transparency enables the analyst to take a look into the interior of a surface ensemble, where many surfaces overlap. Although it does not support exact distinction of individual surfaces, it significantly improves the perception of the surfaces' distribution.

Constant switching between these surface specific views at any time during the analysis supports the trade-off between detail and scalability. The presented views are combined with standard views, such as histogram, scatter plot, and parallel coordinates, by means of brushing techniques as well as focus+context visualization, to benefit from the capabilities of a coordinated multiple views system.

We evaluate the proposed methodology by means of a case study, which involves multiple-run simulation of gear contact stresses for automotive domain. Simulation is used to establish a gear design that satisfies the target requirements relating to safety factors, noise, minimized gear failure, and increased service life, for example. Traditional approaches require the modification of tooth profiles in iterative cycles, which consist of simulation, gear contact analysis, and modification. In contrast to that, the presented methodology is based on a large number of pre-computed simulation runs (e.g. 900), where each run is executed with a different gear design setting as input. This approach enables a simultaneous exploration and investigation of gear contact stresses resulting from various gear designs, without the need to perform simulation over and over again. Compared to conventional approaches, domain experts highly appreciated the novel techniques. They stated that the novel methodology simplifies and at the same time deepens the understanding of inter-dependencies between gear design and gear stresses. According to domain experts, the opportunity to gain insights into the impact of changes in gear geometry on resulting gear contact forces exceeds the possibilities of state-of-the-art tools. For tasks that can also be solved using conventional tools, the speed-up of analysis is estimated to be at least an order of magnitude. Other tasks are not supported by a traditional workflow, which makes it hard to establish a quantitative speed-up estimation.

Surface ensembles occur in various domains like meteorology, engineering, or lighting design, which makes the newly proposed methodology widely applicable. In this context, the methodology may be beneficial, as it provides a powerful collection of tools for data analysis and exploration from different perspectives. The approach can also be attuned to meet the analysis tasks relating to a specific application.

6. Future Work

The interactive visual analysis approach presented in this thesis supports exploration of multi-dimensional data containing data surfaces as atomic unit, which emerge from advanced multiple-run simulation. Proven techniques are combined with novel visualizations to enable insights into the data from different perspectives, which would not have been possible otherwise. However, there are still limitations of the presented visualizations, which need to be taken into account. The approach also offers a lot of potential to be extended in various directions, allowing it to be utilized in several application domains.

For depicting multiple data surfaces simultaneously, we propose a linear transparency transfer function to cope with occlusion. However, this approach reaches its limits if the number of displayed surfaces exceeds a certain amount, with the consequence that data surfaces are still hard to distinguish. The perception of shapes, which pose an important source of information, might be improved by applying advanced methods from the field of layered surface texturing.

Evaluation of the presented approach by domain experts only considers a rather simple and idealistic model of a gearing setup. As a consequence, the evaluation results cannot be used to make a reliable statement about the method's benefit for realistic application cases. Thus, there is a need for reviewing the proposed methodology when applied to more complex gearing systems. In addition, domain experts stated that it would be helpful to be able to simultaneously investigate multiple physical quantities, such as normal force, sliding velocity, and others, resulting in a deeper understanding of gear contact phenomena. To support this issue, the proposed techniques need to be further developed towards comparison of several surface ensembles. Considering applications other than gears for evaluation would allow for a more general assessment of the approach's practicability.

The case study presented in this thesis furthermore solely confines to spur gears. However, depending on the target application, other types of gear, such as helical gears,

are commonly used in practice as well. Extending the proposed methodology to other types of gears would be a valuable achievement. When considering helical gears, we suggest to use skewed heat maps to cope with differently oriented tooth flanks.

In various application domains, frequency-domain analysis is widely used to obtain new insights into the underlying data. Regarding gearing systems, this domain is especially suitable for noise and vibration analysis. This issue can be addressed by transforming surfaces into frequency-domain as follows: (1) Extract multiple curves from the surface by creating a finite number of cuts parallel to one axis. (2) Transform the curves to frequency-domain. One original curve is represented by exactly one curve in frequency-domain. (3) Compose the curves in frequency-domain to a surface again, which can then be analyzed using the presented interactive visual analysis approach.

Regarding the data set originating from ensemble simulation of gears, frequency-domain analysis could also be used for aggregation along teeth for a fixed simulation run. As we have time series summarizing gear characteristics (e.g. total normal force) along tooth flanks, Fourier transform decomposes them into multiple curves representing the constituent frequencies. In turn, these curves can be composed to a surface representing the feature in frequency-domain.

Aggregating surfaces by deriving time series or scalar values currently follows a rather simple scheme. Derivated values are obtained by computing commonly used statistical measures such as maximum, for example. Additionally offering more advanced statistics for data derivation could enable more informative surface aggregation.

Interactive visual analysis of the given ensemble simulation data is limited in such a way that it is based on ideal operating conditions and only left flanks of pinion teeth are considered. Imperfections like vibrations lead to contacts that jump between left and right flank of a certain tooth, due to closing backlash between driving and driven flank. When considering both flanks per tooth, advanced techniques have to be established to cope with such difficult conditions. Surfaces representing right and left flank of a tooth respectively could be linked in order to keep track of alternating contacts.

A. Appendix

Pinion				Gear		
Press. Angle Correction	Barrelling	Tip Relief	Crowning	Press. Angle Correction	Barrelling	Crowning
z1_C_Halpha	z1_C_alpha	z1_C_alpha_a	z1_C_beta	z2_C_Halpha	z2_C_alpha	z2_C_beta
9.1406	16.0938	0.0977	8.8281	1.0156	4.8438	0.3906
4.1406	6.0938	0.0477	3.8281	6.0156	14.8438	5.3906
1.6406	1.0938	0.0727	1.3281	3.5156	9.8438	2.8906
6.6406	11.0938	0.0227	6.3281	8.5156	19.8438	7.8906
7.2656	4.8438	0.0414	5.7031	7.8906	8.5938	1.0156
2.2656	14.8438	0.0914	0.7031	2.8906	18.5938	6.0156
4.7656	19.8438	0.0164	3.2031	5.3906	3.5938	3.5156
9.7656	9.8438	0.0664	8.2031	0.3906	13.5938	8.5156
8.5156	12.3438	0.0289	4.4531	1.6406	16.0938	2.2656
3.5156	2.3438	0.0789	9.4531	6.6406	6.0938	7.2656
1.0156	7.3438	0.0039	6.9531	4.1406	11.0938	4.7656
6.0156	17.3438	0.0539	1.9531	9.1406	1.0938	9.7656
5.7031	14.2188	0.0695	2.8906	6.3281	5.4688	8.8281
0.7031	4.2188	0.0195	7.8906	1.3281	15.4688	3.8281
3.2031	9.2188	0.0945	5.3906	8.8281	0.4688	6.3281
8.2031	19.2188	0.0445	0.3906	3.8281	10.4688	1.3281
9.4531	1.7188	0.057	6.6406	2.5781	17.9688	7.5781
4.4531	11.7188	0.007	1.6406	7.5781	7.9688	2.5781
1.9531	16.7188	0.082	4.1406	0.0781	12.9688	5.0781
6.9531	6.7188	0.032	9.1406	5.0781	2.9688	0.0781
6.3281	17.9688	0.0008	8.5156	5.7031	14.2188	8.2031
1.3281	7.9688	0.0508	3.5156	0.7031	4.2188	3.2031
3.8281	2.9688	0.0258	1.0156	8.2031	19.2188	5.7031
8.8281	12.9688	0.0758	6.0156	3.2031	9.2188	0.7031
7.5781	5.4688	0.0133	2.2656	4.4531	1.7188	9.4531

Pinion				Gear		
Press. Angle Correction	Barrelling	Tip Relief	Crowning	Press. Angle Correction	Barrelling	Crowning
z1_C_Halpha	z1_C_alpha	z1_C_alpha_a	z1_C_beta	z2_C_Halpha	z2_C_alpha	z2_C_beta
2.5781	15.4688	0.0633	7.2656	9.4531	11.7188	4.4531
0.0781	10.4688	0.0383	9.7656	1.9531	6.7188	6.9531
5.0781	0.4688	0.0883	4.7656	6.9531	16.7188	1.9531
5.1172	5.7031	0.0441	9.6484	9.1797	19.7656	2.3047
0.1172	15.7031	0.0941	4.6484	4.1797	9.7656	7.3047
2.6172	10.7031	0.0191	2.1484	6.6797	14.7656	4.8047
7.6172	0.7031	0.0691	7.1484	1.6797	4.7656	9.8047
8.8672	18.2031	0.0316	0.8984	0.4297	7.2656	1.0547
3.8672	8.2031	0.0816	5.8984	5.4297	17.2656	6.0547
1.3672	3.2031	0.0066	8.3984	2.9297	2.2656	3.5547
6.3672	13.2031	0.0566	3.3984	7.9297	12.2656	8.5547
6.9922	1.9531	0.0754	4.0234	8.5547	1.0156	0.4297
1.9922	11.9531	0.0254	9.0234	3.5547	11.0156	5.4297
4.4922	16.9531	0.0504	6.5234	6.0547	6.0156	2.9297
9.4922	6.9531	0.0004	1.5234	1.0547	16.0156	7.9297
8.2422	14.4531	0.0879	5.2734	2.3047	13.5156	1.6797
3.2422	4.4531	0.0379	0.2734	7.3047	3.5156	6.6797
0.7422	9.4531	0.0629	2.7734	4.8047	18.5156	4.1797
5.7422	19.4531	0.0129	7.7734	9.8047	8.5156	9.1797
6.0547	11.3281	0.0098	7.4609	6.9922	2.8906	9.4922
1.0547	1.3281	0.0598	2.4609	1.9922	12.8906	4.4922
3.5547	6.3281	0.0348	4.9609	9.4922	7.8906	6.9922
8.5547	16.3281	0.0848	9.9609	4.4922	17.8906	1.9922
9.8047	3.8281	0.0223	3.7109	3.2422	10.3906	8.2422
4.8047	13.8281	0.0723	8.7109	8.2422	0.3906	3.2422
2.3047	18.8281	0.0473	6.2109	0.7422	15.3906	5.7422
7.3047	8.8281	0.0973	1.2109	5.7422	5.3906	0.7422
6.6797	15.0781	0.066	1.8359	5.1172	16.6406	7.6172
1.6797	5.0781	0.016	6.8359	0.1172	6.6406	2.6172
4.1797	0.0781	0.091	9.3359	7.6172	11.6406	5.1172
9.1797	10.0781	0.041	4.3359	2.6172	1.6406	0.1172
7.9297	7.5781	0.0535	8.0859	3.8672	9.1406	8.8672
2.9297	17.5781	0.0035	3.0859	8.8672	19.1406	3.8672
0.4297	12.5781	0.0785	0.5859	1.3672	4.1406	6.3672
5.4297	2.5781	0.0285	5.5859	6.3672	14.1406	1.3672
5.5859	16.6406	0.0895	2.6172	3.0859	6.3281	6.8359
0.5859	6.6406	0.0395	7.6172	8.0859	16.3281	1.8359
3.0859	1.6406	0.0645	5.1172	0.5859	1.3281	9.3359
8.0859	11.6406	0.0145	0.1172	5.5859	11.3281	4.3359
9.3359	9.1406	0.077	6.3672	6.8359	18.8281	5.5859
4.3359	19.1406	0.027	1.3672	1.8359	8.8281	0.5859
1.8359	14.1406	0.052	3.8672	9.3359	13.8281	8.0859

Pinion				Gear		
Press. Angle Correction	Barrelling	Tip Relief	Crowning	Press. Angle Correction	Barrelling	Crowning
z1_C_Halpha	z1_C_alpha	z1_C_alpha_a	z1_C_beta	z2_C_Halpha	z2_C_alpha	z2_C_beta
6.8359	4.1406	0.002	8.8672	4.3359	3.8281	3.0859
7.4609	10.3906	0.0332	8.2422	4.9609	12.5781	6.2109
2.4609	0.3906	0.0832	3.2422	9.9609	2.5781	1.2109
4.9609	5.3906	0.0082	0.7422	2.4609	17.5781	8.7109
9.9609	15.3906	0.0582	5.7422	7.4609	7.5781	3.7109
8.7109	2.8906	0.0457	1.9922	6.2109	0.0781	7.4609
3.7109	12.8906	0.0957	6.9922	1.2109	10.0781	2.4609
1.2109	17.8906	0.0207	9.4922	8.7109	5.0781	9.9609
6.2109	7.8906	0.0707	4.4922	3.7109	15.0781	4.9609
5.8984	1.0156	0.0551	0.4297	0.8984	10.7031	4.0234
0.8984	11.0156	0.0051	5.4297	5.8984	0.7031	9.0234
3.3984	16.0156	0.0801	7.9297	3.3984	15.7031	1.5234
8.3984	6.0156	0.0301	2.9297	8.3984	5.7031	6.5234
9.6484	13.5156	0.0676	9.1797	9.6484	3.2031	2.7734
4.6484	3.5156	0.0176	4.1797	4.6484	13.2031	7.7734
2.1484	8.5156	0.0926	1.6797	7.1484	8.2031	0.2734
7.1484	18.5156	0.0426	6.6797	2.1484	18.2031	5.2734
6.5234	7.2656	0.0238	6.0547	1.5234	9.4531	3.3984
1.5234	17.2656	0.0738	1.0547	6.5234	19.4531	8.3984
4.0234	12.2656	0.0488	3.5547	4.0234	4.4531	0.8984
9.0234	2.2656	0.0988	8.5547	9.0234	14.4531	5.8984
7.7734	19.7656	0.0113	4.8047	7.7734	16.9531	4.6484
2.7734	9.7656	0.0613	9.8047	2.7734	6.9531	9.6484
0.2734	4.7656	0.0363	7.3047	5.2734	11.9531	2.1484
5.2734	14.7656	0.0863	2.3047	0.2734	1.9531	7.1484
5.1953	4.6094	0.0746	7.5391	2.0703	6.1719	0.1953
0.1953	14.6094	0.0246	2.5391	7.0703	16.1719	5.1953
2.6953	19.6094	0.0996	0.0391	4.5703	1.1719	2.6953
7.6953	9.6094	0.0496	5.0391	9.5703	11.1719	7.6953
8.9453	12.1094	0.0621	1.2891	8.3203	18.6719	1.4453
3.9453	2.1094	0.0121	6.2891	3.3203	8.6719	6.4453
1.4453	7.1094	0.0871	8.7891	5.8203	13.6719	3.9453
6.4453	17.1094	0.0371	3.7891	0.8203	3.6719	8.9453

Table A.1: 100 combinations of seven profile modifications resulting from Sobol sequences. They are used as control parameters for simulation of a gearing system.

Bibliography

- [ASc94] §21 Abs. 2, §22 Abs. 6, ArbeitnehmerInnenschutzgesetz BGBl 450/1994. https://www.ris.bka.gv.at/Dokumente/BgblPdf/1994_450_0/1994_450_0.pdf, 1994. Austrian legislature. Accessed: 2016-04-04. (cited on Page 15)
- [ASt98] §§ 25 and 29, Arbeitsstättenverordnung BGBl II 368/1998. https://www.ris.bka.gv.at/Dokumente/BgblPdf/1998_368_2/1998_368_2.pdf, 1998. Austrian legislature. Accessed: 2016-04-04. (cited on Page 15)
- [AWH⁺12] Oluwafemi S. Alabi, Xunlei Wu, Jonathan M. Harter, Madhura Phadke, Lifford Pinto, Hannah Petersen, Steffen Bass, Michael Keifer, Sharon Zhong, Christopher G. Healey, and Russell M. Taylor. Comparative Visualization of Ensembles Using Ensemble Surface Slicing. In *Proceedings of Visualization and Data Analytics (VDA)*, volume 8294, pages 1–12, 2012. (cited on Page 4)
- [Bai09] Alethea S. Bair. *Optimization of Single and Layered Surface Texturing*. PhD thesis, Texas A&M University, United States, 2009. (cited on Page 36)
- [Ben88] Yoav Benjamini. Opening the Box of a Boxplot. *The American Statistician*, 42(4):257–262, 1988. (cited on Page 22)
- [BSM⁺13] Steven Bergner, Michael Sedlmair, Torsten Möller, Sareh Nabi Abdolyousefi, and Ahmed Saad. ParaGlide: Interactive Parameter Space Partitioning for Computer Simulations. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1499–1512, 2013. (cited on Page 1 and 8)
- [CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999. (cited on Page 11, 12, and 14)
- [Col87] J. R. Colbourne. *The Geometry of Involute Gears*. Springer-Verlag New York, 1987. (cited on Page 39, 40, and 41)
- [Dra88] Raymond J. Drago. *Fundamentals of Gear Design*. Butterworth-Heinemann, 1988. (cited on Page 39 and 40)

- [Fis95] Paul A. Fishwick. *Simulation Model Design and Execution: Building Digital Worlds*. Prentice Hall PTR, 1995. (cited on Page 1)
- [Flo00] Anders Flodin. *Wear of Spur and Helical Gears*. PhD thesis, Royal Institute of Technology, Sweden, 2000. (cited on Page 40 and 42)
- [GLG⁺13] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, 2013. (cited on Page 8)
- [Gmb] AVL LIST GmbH. AVL EXCITE Power Unit. https://www.avl.com/excite/-/asset_publisher/gYjUpY19vEA8/content/avl-excite-power-unit. Accessed: 2016-04-04. (cited on Page 43)
- [Gmb12] AVL LIST GmbH. *AVL EXCITE Power Unit: User's Guide*, v2011.2 edition, 5 2012. (cited on Page 43)
- [GRW⁺00] Donna L. Gresh, Bernice E. Rogowitz, Raimond L. Winslow, David F. Scolan, and Christina K. Yung. WEAVE: A System for Visually Linking 3-D and Statistical Visualizations, Applied to Cardiac Simulation and Measurement Data. In *Proceedings of the IEEE Conference on Visualization*, pages 489–492. IEEE, 2000. (cited on Page 7)
- [Han15] Jonny Hansen. Gear Contact Simulation on Asperity Level. Master's thesis, Luleå University of Technology, Sweden, June 2015. (cited on Page 40)
- [Hau06] Helwig Hauser. Generalizing focus+context visualization. In *Scientific Visualization: The Visual Extraction of Knowledge from Data*, chapter Generalizing Focus+Context Visualization, pages 305–327. Springer, 2006. (cited on Page 7)
- [HB03] Mark Harrower and Cynthia A. Brewer. ColorBrewer.org: an Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, 40(1):27–37, 2003. (cited on Page 30)
- [HJ11] Charles D. Hansen and Chris R. Johnson. *Visualization Handbook*. Academic Press, 2011. (cited on Page 36)
- [KAF⁺08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual Analytics: Definition, Process, and Challenges*. Springer, 2008. (cited on Page 2)
- [KDP01] David Kao, Jennifer L. Dungan, and Alex Pang. Visualizing 2D Probability Distributions from EOS Satellite Image-Derived Data Sets: A Case Study. In *Proceedings of the IEEE Conference on Visualization*, pages 457–589. IEEE, 2001. (cited on Page 9)

- [Kei01] Daniel A. Keim. Visual Exploration of Large Data Sets. *Communications of the ACM*, 44(8):38–44, 2001. (cited on Page 3 and 6)
- [KLM⁺12] Zoltán Konyha, Alan Lež, Krešimir Matković, Mario Jelović, and Helwig Hauser. Interactive Visual Analysis of Families of Curves Using Data Aggregation and Derivation. In *Proceedings of the International Conference on Knowledge Management and Knowledge Technologies*, page 24. ACM, 2012. (cited on Page 14)
- [KMG⁺06] Zoltán Konyha, Krešimir Matković, Denis Gračanin, Mario Jelović, and Helwig Hauser. Interactive Visual Analysis of Families of Function Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1373 – 1385, 2006. (cited on Page 9 and 38)
- [LS04] Greg W. Larson and Rob Shakespeare. *Rendering with Radiance: the Art and Science of Lighting Visualization*. Booksurge Llc, 2004. (cited on Page 15)
- [MFGH08] Krešimir Matković, Wolfgang Freiler, Denis Gračanin, and Helwig Hauser. ComVis: a Coordinated Multiple Views System for Prototyping New Visualization Technology. In *Proceedings of the International Conference on Information Visualisation (IV)*, pages 215–220. IEEE, 2008. (cited on Page 7, 10, and 37)
- [MGJH15] Krešimir Matković, Denis Gračanin, Mario Jelović, and Helwig Hauser. Interactive Visual Analysis of Large Simulation Ensembles. In *Proceedings of the 2015 Winter Simulation Conference*, pages 517–528. IEEE, 2015. (cited on Page 3)
- [MGKH09] Krešimir Matković, Denis Gračanin, Borislav Klarin, and Helwig Hauser. Interactive Visual Analysis of Complex Scientific Data as Families of Data Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1351–1358, 2009. (cited on Page 10, 11, 21, 32, and 36)
- [MGS⁺15] Krešimir Matković, Denis Gračanin, Rainer Splechtna, Alexandra Diehl, Mai Elshehaly, and Claudio Delrieux. Exploring Trajectory Data Using ComVis CMV Tool - VAST 2015 Mini-Challenge 1. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 167–168. IEEE, 2015. (cited on Page 38)
- [MJJ⁺05] Krešimir Matković, Mario Jelović, Josip Jurić, Zoltán Konyha, and Denis Gračanin. Interactive Visual Analysis and Exploration of Injection Systems Simulations. In *Proceedings of the IEEE Conference on Visualization*, pages 391–398. IEEE, 2005. (cited on Page 8 and 9)
- [MKO⁺08] Philipp Muigg, Johannes Kehrer, Steffen Oeltze, Harald Piringer, Helmut Doleisch, Bernhard Preim, and Helwig Hauser. A Four-level Focus+Context

- Approach to Interactive Visual Analysis of Temporal Features in Large Scientific Data. *Computer Graphics Forum*, 27(3):775–782, 2008. (cited on Page 7)
- [MS00] D.J. Murray-Smith. The Inverse Simulation Approach: a Focused Review of Methods and Applications. *Mathematics and Computers in Simulation*, 53(4):239–247, 2000. (cited on Page 2)
- [NS00] Chris North and Ben Shneiderman. Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 128–135. ACM, 2000. (cited on Page 7)
- [Pot06] Kristin Potter. Methods for Presenting Statistical Information: The Box Plot. In Hans Hagen, Andreas Kerren, and Peter Dannenmann, editors, *Visualization of Large and Unstructured Data Sets*, GI-Edition Lecture Notes in Informatics (LNI) S-4, pages 97–106, 2006. (cited on Page 21)
- [PPBT12] Harald Piringer, Stephan Pajer, Wolfgang Berger, and Heike Teichmann. Comparative Visual Analysis of 2D Function Ensembles. *Computer Graphics Forum*, 31(3):1195–1204, 2012. (cited on Page 9, 14, and 18)
- [PSMD14] Luciana Padua, Hendrik Schulze, Krešimir Matković, and Claudio Delrieux. Interactive Exploration of Parameter Space in Data Mining: Comprehending the Predictive Quality of Large Decision Tree Collections. *Computers & Graphics*, 41:99–113, 2014. (cited on Page 10)
- [PWB⁺09] Kristin Potter, Andrew Wilson, Peer-Timo Bremer, Dean Williams, Charles Doutriaux, Valerio Pascucci, and Chris R. Johnson. Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 233–240. IEEE, 2009. (cited on Page 9)
- [Rad12] Stephen P. Radzevich. *Theory of Gearing: Kinematics, Geometry, and Synthesis*. CRC Press, 2012. (cited on Page 39)
- [RLS⁺96] Steven F. Roth, Peter Lucas, Jeffrey A. Senn, Christina C. Gomberg, Michael B. Burks, Philip J. Stroffolino, John A. Kolojejchick, and Carolyn Dunmire. Visage: A User Interface Environment for Exploring Information. In *Proceedings IEEE Symposium on Information Visualization*, pages 3–12. IEEE, 1996. (cited on Page 7)
- [Rob07] Jonathan C. Roberts. State of the Art: Coordinated & Multiple Views in Explanatory Visualization. In *Proceedings of the International Conference on Coordinated and Multiple Views in Explanatory Visualization (CMV)*, pages 61–71. IEEE, 2007. (cited on Page 7)

- [SA82] Robert Spence and Mark Apperley. Data Base Navigation: An Office Environment for the Professional. *Behaviour and Information Technology*, 1(1):43–54, 1982. (cited on Page 7)
- [SB92] Manojit Sarkar and Marc Brown. Graphical Fisheye Views of Graphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pages 83–91. ACM, 1992. (cited on Page 7)
- [SB11] John A. Sokolowski and Catherine M. Banks. *Principles of Modeling and Simulation: A Multidisciplinary Approach*. John Wiley & Sons, 2011. (cited on Page 1)
- [SBP⁺12] Martin Sopouch, Christoph Barchanski, Helmut Petrin, Günter Offner, and Josef Haslinger. Simulation of Gearbox Structure Borne Noise Using a Detailed Multi-body Dynamics Gear Contact Model. In *Internationaler VDI-Kongress Getriebe in Fahrzeugen*. VDI Wissensforum GmbH, 2012. (cited on Page 44)
- [SHB⁺14] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014. (cited on Page 6 and 7)
- [Shi67] Eugene E. Shipley. Gear Failures - how to recognize them, what causes them and how to avoid them. *Machine Design*, 39(28):152, 1967. (cited on Page 39)
- [SOL⁺15] Johannes Sorger, Thomas Ortner, Christian Luksch, Michael Schwärzler, Eduard Gröller, and Harald Piringer. LiteVis: Integrated Visualization for Simulation-Based Decision Support in Lighting Design. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):290–299, 2015. (cited on Page 2, 8, and 15)
- [WP09] Andrew T. Wilson and Kristin C. Potter. Toward Visual Analysis of Ensemble Data Sets. In *Proceedings of the 2009 Workshop on Ultrascale Visualization*, pages 48–53. ACM, 2009. (cited on Page 8)
- [ZFB10] Yi Zhang, Susan Finger, and Stephanie Behrens. Introduction to Mechanisms. <http://www.cs.cmu.edu/~rapidproto/mechanisms/chpt7.html#top>, 2010. Carnegie Mellon University, Pennsylvania. Accessed: 2016-04-04. (cited on Page 40)