

Visualisierungsinstitut der Universität Stuttgart

Masterarbeit

Visual Parameters Space Analysis for Architectural Design Space Exploration

Sebastian Grund

Course of Study:	Informatik
Examiner:	Prof. Dr. Michael Sedlmair
Supervisor:	Dr. Quynh Quang Ngo, Dr. Kuno Kurzhals, Moataz Abdellal, Prof. Thomas Wortmann, Max Benjamin Zorn
Commenced:	March 1, 2023
Completed:	October 31, 2023

Abstract

Architects, in their search for building designs, are offered multiple supportive tools by various CAD Software. One such tool is multi-objective optimization (MOO) where a computer proposes possible designs that try to optimize multiple performance functions. But in the past, it was disclosed that the results of such Optimization runs can be hard to interpret and work with correctly. Together with experienced field experts, we analyzed the problem and derived the necessary design requirements for a possible supportive interface. Working closely with these experts we then developed an interactive visualization prototype that aims to help architects with their MOO design approach. To support the exploration of the multi-dimensional parameter space and the multi-dimensional objective space the presented prototype utilizes multiple coordinated views. The Views include among others a parallel coordinates plot of the data points for correlation detection and two scatter plots displaying each only the dimensional reduced parameter and objective part of the MOO results. Together with automated and manual clustering as well as other interaction capabilities, the tool supports exploration and filtration steps for the MOO design process. Next to the direct feedback of the field experts during the iterative development stages, we also evaluated the prototype in a hands-on expert review where MOO results for two different Models were analyzed. The chosen coding style tries to lower required expenses for the addition of more views should additional requirements or different visualization approaches occur. Our tool aims to be integrated into an existing MOO tool (opossum for Rhino3D) and become the interlinking part between the computer and the architect in their cooperative building design search.

Contents

1	Introduction	9
2	Background	13
2.1	Parametric Approach in Architectural Modelling	13
2.2	Automatic Optimization in Architectural Design	13
2.3	Rhino3D	14
3	Related Work	15
3.1	Visualization of Multivariate Data	15
3.2	Interpretation of Dimensionality Reduction	16
3.3	Performance Map	18
4	Methodology and Requirements	19
4.1	Methodology	19
4.2	Task Analysis and Design Requirements	19
5	Method	21
5.1	Data Formalization	21
5.2	Multiple Coordinated Views	22
5.3	2D Spatial Mapping	23
5.4	Cluster Detection	25
5.5	Cluster Attribution and Distinction	25
5.6	Bijjective Representation	27
5.7	Correlation Detection	28
6	Interface Description	31
6.1	Modularity	31
6.2	Use Case	32
6.3	Implementation	34
7	Evaluation	37
7.1	Procedure and Tasks	37
7.2	Datasets	37
7.3	Findings	39
8	Discussion	43
8.1	Expert Feedback	43
8.2	Grasshopper Integration	45
8.3	Future Work	45

9 Conclusion	47
Bibliography	49

List of Figures

1.1	Screenshots of the CAD-Software Rhino3D and its Grasshopper extension	10
3.1	A tSNE dimensional reduced 2D Mapping with two vector representations of BIOT matrices. Each vector shows the local approximation of the 2D-displacement effect of changes in one dimension. Each vector group shows these displacements for all encoded dimensions.	17
3.2	Performance Map	18
5.1	Two 2D-Cluster-Views showing the parameter Clusters on the left and the Objective Clusters on the right.	24
5.2	Selection and Move Functionality in the 2D-Cluster-View. Both show the same clusters but the Figure 5.2b allows for better analysis of parts of the square cluster.	24
5.3	Box Plot Explanations: A longer or shorter line on the top or the bottom shows heavy or not-so-heavy outliers. A shorter or longer box means that the bulk (middle 50%) of the values are more or less concentrated. And a the median-line gives us information of the distributions inside of this bulk part.	26
5.4	A Box-Plot-Matrix showing multiple Box-Plots. It shows distinctive Features of each data group and allows to differentiate and compare them.	27
5.5	Showing a selection done in the PCP-View by limiting certain dimensions. ('Par rotat.' and 'Opt sh-size') The limits can be seen as Orange ranges on the filtered dimensions.	29
6.1	A overview of the default user interface.	31
6.2	Two examples for Corelations showing in PCPs.	33
6.3	Corse Diagram of MVVM-Concept.	35
7.1	Simple Evaluation Model	38
7.2	Realistic Evaluation Model opened in Rhino/Grasshopper.	38
7.3	Recreations of the interaction steps utilized by the domain expert on the simple models MOO results during the evaluation.	40
7.4	Recreations of the interaction steps utilized by the domain expert on the realistic models MOO results during the evaluation.	42

1 Introduction

Humans have designed their buildings, compromising between multiple goals since prehistoric times and before they had the help of computers. Ancient civilizations managed to erect impressive structures that, if still standing, amaze us today and must have seemed impossible for ordinary people in their time of creation. The survived term “7 wonders of the world” and preserved descriptions and scratches from chronologists reveal the effect some buildings had on individuals throughout the ages. But architecture, the profession, and science behind designing and constructing structures, dates back even further than some would guess. Only recently, Crassard et al. [CAB+23] published their finding of plans to scale showing big animal traps in today’s Jordania estimated to date back to the Stone Ages.

In their design approaches, architects have to compromise between multiple factors. These factors can be very specific and unique for each building endeavor. But it is imaginable that they include construction time and expense limitations next to subjective aesthetic demands and have to be technically feasible to realize. For their designs to successfully become reality, architects include acerbic planning in their design process. Many antique structures that persisted for thousands of years and also some preserved construction plans are proof of their knowledge and capabilities.

The invention and distribution of computers changed the world and fundamentally changed the architectural design process. Today various Computer-Aided-Design-(CAD)Programs are widely spread and utilized to support architects in a multitude of tasks including building design. In 2010 Tornincasa, Di Monaco, et al. [TD+10] summarised the adaptation, changes but also potential future of CAD Software. Although already 10 years old, the summary shows that new ideas and technical inventions often lead to new use cases for CAD Software being imagined and realized. A recent example would be the possible addition of more Virtual Reality (VR) features into the CAD Process. The points mentioned by Wolfartsberger [Wol19] in his analysis, could make the existence of VR capabilities a selling point for CAD software products.

But also already established design methods, like automatic simulation-based optimization, can become the focus of research. The idea is that computers assist in finding designs that, based on simulation results perform well in some evaluation or optimization functions. In 2022 Wortman et al.’s survey [WCW22] about the practice found non-ideal utilization of optimization with practitioners in industry and academia. Concerns were raised over the canonic usage of Genetic Algorithms for optimization. This practice could lead to non-ideal designs being generated and wrong design conclusions being derived from them as described by Wortmann et al. [WCW22]. Their tool Opossum (**OptimizatiOn Solver with SURrogate Models**) [Wor17a] employs a Model-based Optimization Algorithm named RBFOpt which they compared against established methods.

Opossum is an easy-to-use grasshopper plugin for the CAD Software Rhino3d, see Figure 1.1.

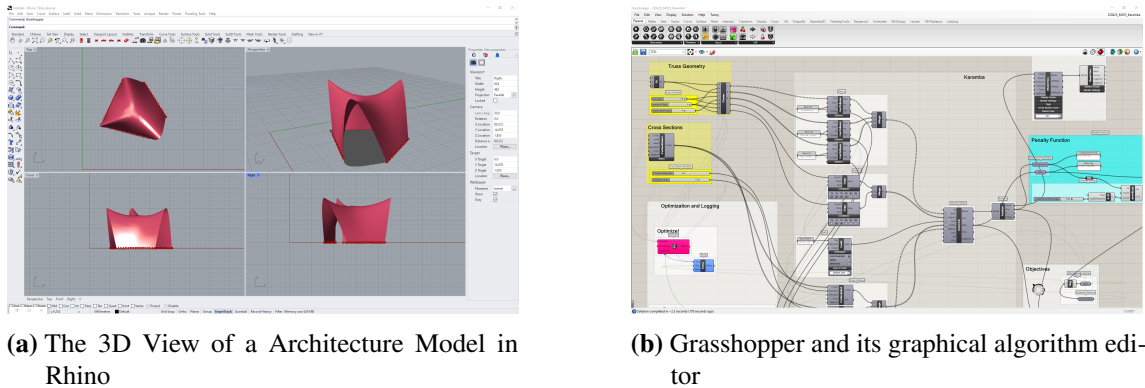


Figure 1.1: Screenshots of the CAD-Software Rhino3D and its Grasshopper extension

Rhino 3D is a 3-D Modelling CAD software not exclusively marked for, but also used by many architects. Grasshopper extends Rhino with a graphical algorithm editor that allows it to easily define calculations and simulations by linking Graphical-Elements. Among other things, it enables Designers to create their 3D models in a way that allows them to easily explore the effects of tweaking parameters (for example, the length of a wall) in the Model. An idea behind this approach is that with multiple Parameters each unique combination of parameter values describes a possible Building-Design. The plugin Opossum adds automatic optimization capabilities for this parametric design approach to the list of grasshopper functionalities. It searches for sets of parameters that perform well in one or many user-defined optimization functions. After calculations, the optimizer offers a list of parameter-value sets (or architectural parameter space) and their calculated/simulated optimization function's performances.

However, to extend the usefulness of the optimization results over those of the simple list, a different interface is needed. For the single-objective use case of Opossum, such an interface was already integrated in the form of Wortmanns performance maps [Wor17b]. It offers a star coordinate mapping of the multi-dimensional parameter value sets where the color of singular points shows the performance of the projected design. The expected performance for spaces in between the mapped points is estimated and also visualized in the Plot. Finally, users are able to manually trigger simulation runs of areas in the star coordinate view that they expect to perform well or let the computer test areas that it expects to hold well performing designs. For the empty space between points the visualization estimates the expected performances by using a surrogate model on interpolated parameters sets considering the surrounding simulated points.

There are known problems for Multi-Objective Optimization (MOO) in the architecture design process, e.g., the much higher complexity and calculation costs compared to Single Objective Optimizations (SOO), or the usage of non-ideal algorithms that fail to produce good parameter vectors. But as determined by Wortmann and Fischer [WF20] in their critical survey on MOO there are also cases where MOO despite its challenges might be beneficial. However, to the best of our knowledge, an interface that provides similar features for MOO results to what performance map allows for SOO ones is still missing. Such a tool should allow for the exploration of the design space (architectural parameter space) and enable human-computer (the optimizer) cooperation.

Every bidirectional communication between a computer and a human requires a form of interface for the human to interact with. Inventing and designing such interactive Interfaces to support them in solving various tasks with computers is a long-known field of research in the Visualization community. Usually, this involves presenting abstract data in forms that users can understand and interact with intuitively. Design approaches and their techniques are nearly as numerous and diverse as the domains their creation originates from. Since as described above, new use case ideas are constantly imagined or old ones are revisited this area of research is not short of movement despite the years of active research. Recent examples that got new visualizations could be Heberle et al.'s invention [HZS+23] to explore artificial intelligence attribute scores on molecule parts in chemistry or also Anand and Rodriguez Lopez's interactive visualization [AR22] for working with chromosome data.

Inspired by the aforementioned work, we aim to provide such an interface that assists architects in co-working with the optimizer and the CAD-Programm in general in finding a valid Building-Design. On a broad level, we work with the presentation and exploration of Multi-Dimensional Points composed of a Multi-Dimensional Parameter-Value-Set and a Multi-Dimensional Objective-Value-Set. Through interviews with field experts, we established tasks that such a possible solution should support users in solving. Users should be able to:

- filter designs based on objective and subjective criteria,
- gain a better understanding of the relation of optimization values to each other,
- gain a better understanding of how different parameters affect different optimization values,
- narrow down the search room for further optimization steps.

In aiming to help them with these requirements we followed the idea to offer a split but linked multi-view on their data. Datapoints should be represented in these views visualizing different aspects and allowing the users via intuitive interactions to explore them based on different intentions. A split design with linked visualizations was a very early choice. The idea was to include separate Projections for the Parameter-Value-Set and the Objective-Value-Set of Optimization Results next to combined ones. We anticipated that building designs with similar parameters might share similar looks or design ideas. Similar Performances in the Objective Functions could help in the evaluation and filtering of the dataset. An automatic clustering based on Parameters as well as one based on Objective-Performance was also included expected to further ease the handling of even big data sets. We also tried to offer functionalities that allow the exploration of correlations between value scales (Parameter-Values and Objective-Performance-Values) for the whole dataset as well as only for selected Groups of Designs. The implemented interface is tested by a field expert experienced in working with Opossums. The expert attempted to solve the above-mentioned tasks with our prototype. With the future-planned back channel to the optimizer, we hope to have helped in making MOO a more valid use case for Opossum.

Opossum and Performance Map are actively developed by members of the Institute for Computational Design and Construction at the university of Stuttgart. Their source code is planned to be released at a future point to allow for community development. But the software is already provided for free and easy usage by every grasshopper user. Because of this and because we also support the idea behind open source code we opted for an adaptable coding style that would allow the integration of further views and ideas without a very deep knowledge of the existing code base. After cleaning up the code and fixing the last issues mentioned in Chapter 8, we also plan to open-source the code.

In summary, this thesis's contribution can be listed as follows:

- A list of design requirements after conducting interviews with domain experts for designing a visualization interface for the Multi-Objective Optimization in the architecture design process,
- A visualization interface that aids the visual architectural parameter space analysis and the architecture Multi-Objective Optimization design process,
- An expert study with one domain expert, evaluating the interface,
- The open source code of our implementation, providing a framework so developers can adapt new visualization approaches in the future.

2 Background

In this chapter, we describe the technologies and their concepts that our interface should be integrated into. We also try to elaborate more on what problems we are trying to solve with it.

2.1 Parametric Approach in Architectural Modelling

A parametric approach in Architectural Modelling describes the method to create models that depend on parameter values [Sch07]. This creates a condition where every unique set of these parameter values can define a possible Model Design for the designers. Architects can now manually explore how changes in parameters affect the final design. An example could be to have a variable ceiling height in a building design and then look at how different ceiling heights would affect the overall building. The search for an explicit building design turns into one for a parameter-values-set that describes this design. This narrows down the search Space for the Design Problem tremendously and makes human-computer-co-working approaches more feasible.

2.2 Automatic Optimization in Architectural Design

With the above-explained parameter-based modeling approach and the definition of one or many performance Functions, the design search can be regarded as an optimization problem. Various optimization algorithms can be employed to find a well-performing parameter value set. These optimization functions can be arbitrarily complicated, spanning from simple arithmetical calculations to full-on simulations of the described model. But Architectural design decisions in general also contain a subjective aesthetic factor that is difficult to describe in mathematical form. Because of this a Computer running optimization over the parameter space is only a helping tool in the human-computer-co-working process to decide on a final design.

Optimizations can be categorized into two groups Single Objective Optimization (SOO), describing the Optimization for a single Objective, and Multi Objective Optimization (MOO). SOO makes it easier for algorithms to find good solutions and an objective ranking of parameter-value-sets based on their performance is possible. Multi-Objective Optimization offers more freedom but also complicates the search tremendously. Because objectives can contradict each other clear “best performing” results could not exist. In these cases, a decision for a good balance in these performance Values has to be found by the human architect. If a scaling that represents this balance can be applied before running the optimization, it reduces the complexity of the search and benefits the quality of the results. But if relations between Objectives are unknown MOO can be a good helper in analyzing them [WF20].

2.3 Rhino3D

Rhino 3D is a commercial 3D Modeling Programm marketed for and used by engineers and fabricators to support them in Modelling their Product designs. (<https://www.rhino3d.com/>) While not being the market leader in the field, architects in both academia and industry use Rhino3D for various CAD tasks. A property that sets it apart from other market members is its ability to work precisely with geometric entities that are defined by mathematical equations rather than polygons.

2.3.1 Grasshopper

Grasshopper extends Rhino with a graphical algorithm editor that allows it to easily define calculations and simulations by linking Graphical-Elements. Grasshopper also allows for the development and inclusion of 3rd Party developed plugins. This opens an opportunity for new ideas to be tested out or for special requirements to be integrated into modeling pipelines. A lengthy list of closed and open-source plugins for various functionalities are offered by and for the grasshopper community. Grasshopper allows the intuitive creation of parametric Design-Models

2.3.2 Opossum

Opossum is a Grasshopper Plugin that offers Optimization Capabilities [Wor17a]. It can be included in Grasshopper Projects as an additional Block. To define the search space the block gets connected with a set of Grashopper-Parameters. The Objective Calculations are also done in Grasshopper and their results are then linked with the opossum-Element. Opossum offers multiple Algorithms for its search and can try to minimize or maximize the attached objectives in its search. If started, the opossum triggers parameter changes and the calculation of the objective values in his search for good-performing parameter-value sets. All explored parameter-value sets and their results are collected in a list for further inspection. If SOO was performed they can also be inspected in a connectable view-Node that employs the PerformanceMap interface for further interaction with the user [Wor17b]. For MOO results, a visualization apart from the simple list is missing. Results in their list form are, without further tooling very hard to interpret for humans. To make working with Opossums MOO capabilities feasible for architects we aim to design a visualization that enables them to work with the result.

In talking with field experts we determined tasks that they imagined could be fulfilled by working with an intuitive interface for the provided simulation results.

3 Related Work

At an abstraction level, the resulting data from the architectural Opossum MOO simulation runs are simply high-dimensional data points. And building an interface for communication between architects and Opossums can therefore be classified into the field of visual high-dimensional parameter space analysis. In this chapter, we convey some related work of visualization of multivariate data, interpretation of Dimensionality Reduction (for visual parameter space analysis), and visualization work for architectural designs.

3.1 Visualization of Multivariate Data

3.1.1 Parallel Coordinates Plot (PCP)

Parallel coordinate plots are a bijective visualization method to present Multi-dimensional data points in a 2D plane. Every Dimension is presented as an independent, parallel printed axis. The multi-dimensional Datapoints each get encoded as lines that connect this axis. A line crosses the axis at the corresponding height of the data point value for this dimension. A deeper discussion of PCP can be found in Heinrich and Weiskopfs Article [HW13].

3.1.2 Scatter Plot Matrix (SPLOM)

Scatter Plot Matrices are another 2D plane visualization method for Mutli-dimensional data points. Given a n dimensional data space, Multiple Scatter Plot charts are arranged in a square configuration with side length n . Each Scatter Plots Axis at position (a, b) is now describing the a th and the b th dimension on its axis. The multi dimensional data points are now mapped into a scatter plot point for each of the scatter plots. The result is a matrix of Cartesian charts where correlations of the depicted data set can be easily read. This reason lead to the wide utilization of SPLOMs in many cases of visual parameter space analysis. (See Elmqvist et al. [EDF08]) A limiting factor of SPLOMs are their quadratic space requirements in relation to the dimensionality of the explored data set. A high number of dimensions might require more interactive arrangement of the Scatter Plots.

3.1.3 Star Coordinates

Star Coordinates map the mutli dimensional data point to points in the 2D map by a combination of uniform vectors. (See for example Kandogan [Kan00]) All dimensional values of the data points are normalized and get mapped into the range $[0 - 1]$. Next each dimension receives a 2D vector of uniform length and originating from the same point with same spaced angles. Now each data

point is drawn at the end of a factorized by its components combination of these vectors. This representation has multiple advantages, for example can viewers easily assume where points would move in the 2D plane if certain changes of its values would occur. Disadvantageous is that the mapping method is surjective, meaning multiple data points can map to the same spot in the 2D plane. The easiest example would be the two points that after the normalization step have either 0 or 1 in all dimensions. Because the combination of all uni-length vectors cancels out and the factorizing with 0 also results in no displacement both points are mapped to the origin of the plot.

3.2 Interpretation of Dimensionality Reduction

3.2.1 t-distributed stochastic neighbor embedding (t-SNE)

t-SNE is a dimension reduction method projecting high dimensional data points into ones that are of lower dimensionality. (See for example Wattenberg et al. [WVJ16]) It begins by applying random values for the lower dimensions and then iteratively changes these values. It tries to find a mapping that reduces distances between similar and increases distances between not-so-similar data points. This similarity assumption is calculated based on the distances between data points in the higher dimensional state. The dimension-reduced data can then be used to search for clusters since similar data will be clustered together. But because of its definition the low dimensional t-SNE results can't give any insight into concrete values in the original high-dimensional Data and is also producing undeterministic results. Often two independent runs of tSNE over the same data-set produce similar looking results that seem different only in their rotation.

3.2.2 Best Interpretable Orthogonal Transformation (BIOT)

BIOT is a generalized method of Best Interpretable Rotation (BIR) and tries to find local mapping estimations in non linear dimension reduced embeddings of multi dimensional data points. Given both the high dimensional points $x \in \mathbb{R}^n$ and their embedded representations in the dimensional reduced dataset $\mathbf{dr}(x) \in \mathbb{R}^m$

$$\mathbf{dr}: \mathbb{R}^n \hookrightarrow \mathbb{R}^m$$

BIOT tries to guess a linear mapping $m \in \mathbb{R}^{n \times m}$ that would explain the given representation locally.

$$\mathbf{biot}: \mathbb{R}^n \hookrightarrow \mathbb{R}^m$$

$$x \mapsto m \times x \text{ where locally}$$

$$\mathbf{biot}(x) \approx \mathbf{dr}(x)$$

It then tries to reduce errors in this mapping m over multiple iterative steps.

While logically such linear mapping can't explain the whole dataset in the non linear embedding it can provide a good estimation for local clusters. When visualized and explained this linear mapping can then be used to allow viewers to assume where points would move in the embedding if certain dimensional values would change. (This provides a local explainable similar to the global one explained in Section 3.1.3.)

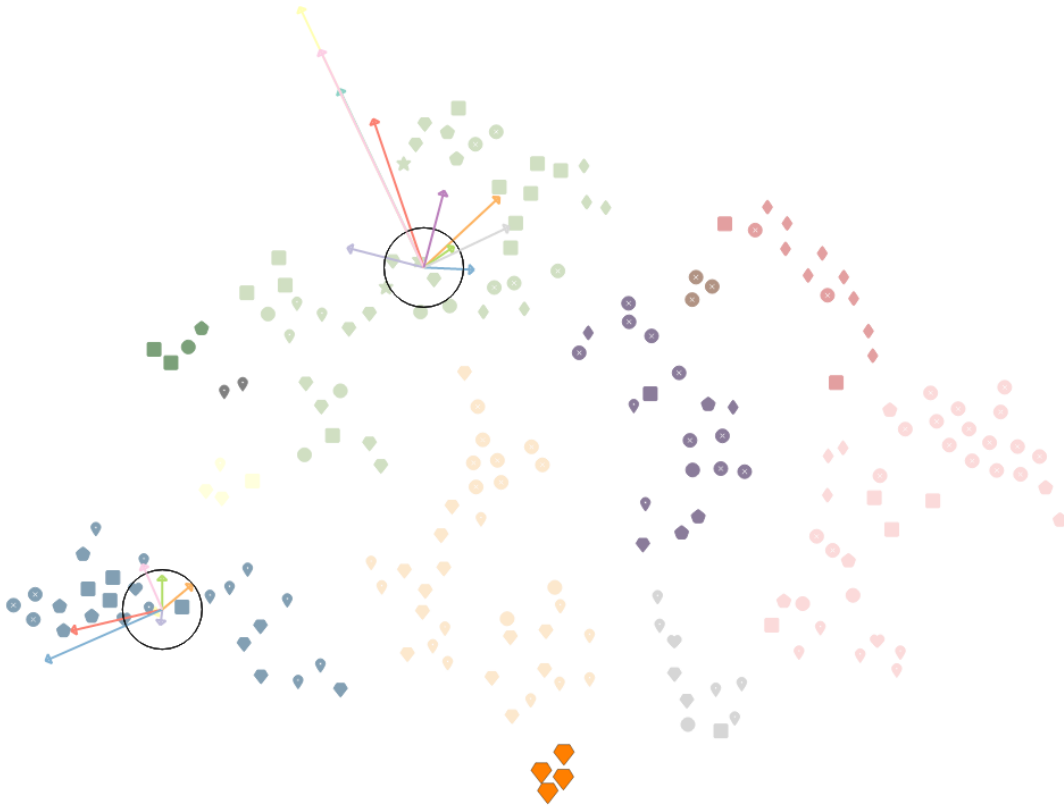


Figure 3.1: A tSNE dimensional reduced 2D Mapping with two vector representations of BIOT matrices. Each vector shows the local approximation of the 2D-displacement effect of changes in one dimension. Each vector group shows these displacements for all encoded dimensions.

Bibal et al. [BMSF21] visualize the embedding of their invention as two rotated perpendicular axes with interactively changeable colored markings of how much a certain dimension is effecting the direction of these axes. For our approach we wanted to offer a visualization of the biot matrix m for each local cluster where the effect of the dimension on the 2D mapping is visualized by different vectors. A screenshot of this representation can be seen in Figure 3.1.

Similar to how Bibal et al. encode more or less impact on the local displacement by coloring more or less of the shown axes, we show this with longer or less long vectors. But we also show the orientation. When two vectors have only a small angular difference to each other this means they

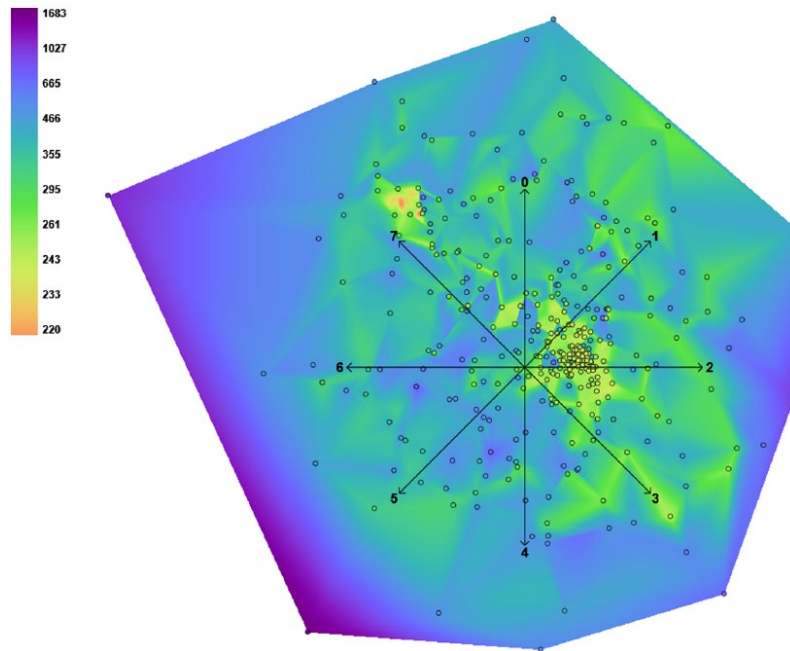


Figure 3.2: Performance Map Visualization of a Design-Space. [Wor17b]

effect the local displacement in a similar way. Because t-SNE groups similar points together this can allow for the assumption that when analyzed local points in the direction of the vectors have rising values for both dimensions therefore a correlation between both of them is more probable. And if the vectors point in opposite directions this allows for an assumption of a negative correlation.

We saw promising results with this method but in review meetings we were made aware that the multi vector visualization is not intuitively understood. So we had to postpone a inclusion for now until we find a more intuitive visual representation of the BIOT matrix.

3.3 Performance Map

In his Article [Wor17b] Wortmann describes a new Visual-Method to explore a multi dimensional parameter space evaluated by a single optimization Method. He presents the simulated points in Star Coordinate Plot and colors them based on their performance. The Spaces in between point get colored based in expected performance of interpolated parameter values of the surrounding concrete simulation results. With this Visualization well-performing Parameter-Sets are easily distinguishable to a Designer. And the predicted performance Areas can be used to guide the optimizer into further exploration.

Compared to our Contribution Performance Map only works for SOO results. But the estimation of performance values is a feature we do not provide at the current state. It is imaginable to include ideas of Performance map into the our work in the future. Multiple Performance Maps next to each other would provide architects with additional hints where to search with the next simulation run. T6.

4 Methodology and Requirements

We provide in this chapter the principles that we based on throughout our study and designing process. The chapter also includes our contribution to task analysis and design requirements for Opossum MOO interface designing.

4.1 Methodology

We followed design studies steps suggested by Sedlmair et al. [SMM12]. In particular, we worked alongside domain experts to perform the task analysis, define design requirements, and develop our design. This chapter presents the derived tasks and the design requirements we defined before we went into design development.

4.2 Task Analysis and Design Requirements

In interviews with two field experts, we got a deeper insight into the current state. The interviewed experts were:

- Expert one is a Tenure Track Professor for Architectural Computing with 15 years of experience in industry and academia.
- Expert two is a Research Associate at an architectural institute with 5 years of experience, working at the ICD Institute and part of the "Computing in Architecture" Working Group of the university of Stuttgart. His research field is the utilization of interactive interfaces in the design process. He studied Architecture and is also part of the opossum development team.

For some design undertakings, MOO can be a very promising tool for human-computer-co-working. The results of these Optimization runs can provide possible knowledge about correlations and also contain the design an architect is deciding on in the end. However, the current list-representation complicates any possibility for knowledge gain or systematic analysis of the results. Without a more elaborated way to work with the results, MOO in its current state seems not feasible for Opossum.

In the interviews discussing the requirements we determined the following domain application tasks that a new interface should strive to support users in working on:

- T 1** filtering designs based on objective performances. Not knowing what balance of the different performance values is acceptable a manual narrowing of the displayed objective space has to be achievable.

- T 2** filtering designs based on subjective artistic criteria. Architecture includes a artistic aspect and architects need to be able to remove building designs that are subjectively not viable. This can be seen as limiting of the parameter space, because only the parameter values define the look of the building.
- T 3** correlation detection between multiple optimization values. Some Objectives share correlations that architects do not know before the simulation. It is a valuable information that certain objectives might contradict each other with the in the current design model. This also helps with managing expectations for “perfekt” designs.
- T 4** correlation detection between parameters and optimization dimensions. Because the Parameter values define the appearance this is a necessary step to link certain visual features with certain objective expectations. It also allows for a co-working between the human and the computer/optimizer because the architects can try to reduce the parameter search space when certain correlations are evident.
- T 5** narrowing down the search room for further simulation runs. Solutions need to provide a communication channel to the optimizer for narrowing down the Parameter search space. A smaller search space improves the likelihood of finding close to or optimal designs in it.
- T 6** triggering the simulation of single, manually specified parameter-value-sets. When analyzing the data points it might be interesting to see the performance of a specific parameter value set. While the values could be manually entered into the Model from the Grasshopper Project the inclusion of such a back channel is preferable to preserve a constant workflow.

For all of these domain Tasks different techniques are thinkable and we discuss the taken implementation paths in Section 5. Our contribution is the analysis of the problem, the implemented functionalities, and finally, the provided prototype that may get integrated into Opossum in the future.

5 Method

To make sure we align with the described requirements, the presented prototype was created in an iterative process with the field experts. Planned features and their implementation were discussed during multiple stages of the development process.

The typical users of our Visualization will be architects in their professional work environment. This allows the assumption of a certain willingness to learn new visualization or interaction methods. Nevertheless, intuitive Design Choices should be preferred where possible to reduce learning phases and prevent misinterpretations.

This chapter presents our main contribution to the interface that aids the architectural parameter space analysis and the Opossum MOO designing process. We start first with the formal description of the simulation data from MOO that we aim to support visual analysis.

5.1 Data Formalization

While possible Parameters in Architecture may include quantitative and qualitative ones we decided with the experts that a mapping to a quantitative scale in \mathbb{R} is always possible. Qualitative Variables can be easily mapped to numbers in \mathbb{N} and therefore in \mathbb{R} by mapping them to the indices of the possible states. We then end up with a formal representation of a Simulation Run like shown in Equation 5.1.

$$(5.1) \quad \mathbf{MOOSim}: \mathbb{R}^m \rightarrow \mathbb{R}^k \\ (p_1, \dots, p_m) \mapsto (o_1, \dots, o_k)$$

Where m is the number of Parameters and k the number of Optimization Functions. From here we can continue to define a single Data-Point as in Equation 5.2.

$$(5.2) \quad \mathbf{Data-Points}: \{\forall x \in \mathbb{R}^{m+k} : \mathbf{MOOSim}(x[1], \dots, x[m]) = (x[m+1], \dots, x[m+k])\}$$

Every Optimization Approach will produce a sub-set of those **Data-Points**. For n MOO Results our whole-input can be defined as in Equation 5.3.

$$(5.3) \quad \mathbf{MOO Results}: \mathbb{R}^{(m+k) \times n} \\ \begin{bmatrix} d_1 \in \mathbf{Data-Points} \\ \vdots \\ d_n \in \mathbf{Data-Points} \end{bmatrix}$$

In the future we also talk about Scales, Scales are equal to a Parameter or Objective Dimensions but only includes values between the minimal and maximal reachable state during the Simulation. For the Parameter Scales, these extreme values are explicitly configured by the user when defining the

model. The extreme values for the optimization scales are implicitly defined by the limits on the parameter scales. But without exploring the whole Parameter Space they are generally unknown and hard to predict. For our visualizations, we utilize the lowest and highest values occurring in simulation results to be the limits. For additional simulation results, these limits have to be changed accordingly.

$$\begin{aligned}
 \mathbf{Scale}(s_{min}, s_{max}) &= \{x \in \mathbb{R} : s_{min} \leq x \leq s_{max}\} \\
 \mathbf{Scales} &= \{\mathbf{Scale}(a, b) : \forall a, b \in \mathbb{R} \wedge a \leq b\} \\
 \mathbf{MOO Results} &= \begin{bmatrix} p_{1,1} \in P_1 & \dots & p_{1,m} \in P_m & o_{1,1} \in O_1 & \dots & o_{1,k} \in O_k \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p_{n,1} \in P_1 & \dots & p_{n,m} \in P_m & o_{n,1} \in O_1 & \dots & o_{n,k} \in O_k \end{bmatrix} \\
 \text{where} & \quad \forall a, b : P_a \in \mathbf{Scales} \wedge O_b \in \mathbf{Scales}
 \end{aligned}$$

5.2 Multiple Coordinated Views

In the beginning we discussed the idea of a single visualization that could fulfill all tasks. But we quickly decided against it because we believed that the described Tasks would greatly benefit from different coordinated visualization approaches. The usage of Multiple views is not new and was for example summarized by Roberts [Rob07].

The reasons for our decision were:

- Filtering in the provided data points can be separated into two evaluation steps for the user. A Ranking based on appearance to enable filtering out not aesthetically pleasing designs (T2). The looks of Models are depending on their parameter values alone. And a Ranking based on the Objective values (T1). In MOO we have multiple objective values with no clear factorization between them so a manual ranking by the user is necessary. To support both rankings an exploration based on either only the design describing parameters (T2) or only the performances in the simulation (T1) is desirable.
- For Correlation-detecting (T3, T4) on the other hand we derived the need to provide yet another visual encoding. But this time not for an encoding on singular data-points but rather on dimensions. Also Task T4 makes it necessary to include dimensions that would not be shown together if we split the exploration view as discussed above.

When discussing the idea of multiple views with the field experts we also got insights into their experiences with such concepts. In their experiences users, given the chance, tend to appreciate creating their personal interface arrangements and liked the freedom of picking their method of solving tasks when offered multiple options. To allow the interactive usage of the multiple views we established two general concepts that all views should integrate into their functionality.

Selection: All utilized Views should if possible allow subsets of data points to be selected. Those selections are global and selection changes affect all views. Each view is meant to highlight the visual elements that represent the selected data points. For our prototype Data-Views, this is done by enlarging them slightly as well as increasing the Opacity of the others. Together this functionality enables the exploration of the data set by selection brushing [BC87; HW13].

Data Groups: Data groups contain a Name and a subset of data points. They can but do not have to configure a color and/or an icon glyph. Configured, they affect how different Views attempt to portray their data group members.

Data Points can be members of multiple data groups. Data Groups can be created by the user and also by the computer. They allow Categorizations like Cluster-Memberships to be visible throughout multiple Views.

An overview of our realization of the interface is shown in Figure 6.1. In the following, we will step by step explain detail our data acquisition, visual encoding choices, and their respective functionalities to fulfill the requirements list.

5.3 2D Spatial Mapping

The filtering tasks (T2,T1) make it necessary for a exploration and evaluation of data-points based on only parts of each points dimension. A automatic ranking is not possible in general. The Objectives can be impossible to factorize into a single quantifiable ranking and the appearance of a building design is a subjective assessment based on the personal preferences of the designer.

To support users in their exploration and rating of the data set we want to provide a projection where the similarity in multiple dimensions of between data points is detectable. We decided on mapping into 2D to allow for more complex similarity relations to show then in 1D but also to avoid the known problems of three or more dimensional visualizations. For such a projection we have to utilize a feature extraction step with a dimension reduction algorithm, to encode our data into something that is possible to be displayed in two dimensions: $f: \mathbb{R}^k \rightarrow \mathbb{R}^2$

As the dimension reduction algorithm we considered multiple options but finally decided on t-SNE (see Section 3.2.1). The reasons for our decision were:

- We are only interested into the similarity between points and do not need any representation of concrete values in the reduced dimensions. This excludes Principal component analysis (PCA) [JC16] and similar approaches.
- The actual distances between points can be very big and misleading (if integer mapped category parameters are used). So we try to avoid the encoding of the actual distances like seen in for example Multi Dimensional Scaling (MDS) [KW78].
- Since we work with optimization results our data might be very dense in some areas and not so dense in others. But small differences in similarities in the denser areas should have an effect. So we also want to avoid strict density preservation.

To simultaneously embed two categorizations into the 2D Scatter plot we utilize Color and the drawn Icon-Glyph for rendered scatter plot points.

Because of the special nature of the data points they consist of two distinguishable multi dimensional part. These parts can be analyzed and categorized separately. While the parameter value Space dedicates the look of a model, the objective space holds its performance.



Figure 5.1: Two 2D-Cluster-Views showing the parameter Clusters on the left and the Objective Clusters on the right.



(a) A selection Polygon in the 2D-Cluster-View. (b) Moved Points in the 2D-Cluster-View.

Figure 5.2: Selection and Move Functionality in the 2D-Cluster-View. Both show the same clusters but the Figure 5.2b allows for better analysis of parts of the square cluster.

The idea is that Users can operate two of these 2D embedded Views, one on the Parameter-Dimensional-Values and one on the Objective-Dimension-Values of the data points. Clustering can be done on both sides and both categorizations are visible to the user.

An example for two t-SNE Scatter plots can be seen in Figure 5.1 Users can now detect whether similar building designs (Parameter-Value Clusters) perform similarly. (Same Colored Elements have many or just one different Icon-Glyph).

To allow for the wanted selection functionality users can draw a polygon that selects the included points. The Polygon can also be moved around, changing the selected points in the process. This allows users to interactively explore and understand the Dimension-Reduced Projections by looking at other views for the distribution of Values of the selected data points.

Users can also move selected Points in the 2D-View to arrange them to their liking. Al while Destroying the automatically generated Dimension-Reduction Results this can help to visually separate or order data points for later more detailed examination. An example for moved points can be found in Figure 5.2b.

5.4 Cluster Detection

To speed up the process of Task T2 and Task T1 we want to provide the user with a automatic clustering based on the features extracted in Section 5.3. T-SNE already provides us with a mapping that groups similar data points. So we need cluster detection algorithm that tries to find the groups t-SNE gathered.

Since we already decided on a 2D t-SNE presentation we only have to detect clusters of points mapped closely together by t-SNE. There are multiple arguments for one or the other clustering algorithm to be made. We tried Centroid-based clustering (k-means) but finally ruled against it because we recognised that linear separable point clouds can't be expected and it might also not be known beforehand how many different clusters a optimization run produces. To prevent the introduction of additional interaction complexity for the user we also temporally decided against the usage of a hierarchical clustering approach and implemented DB-SCAN as a distribution based Cluster Detection Algorithm for our first version.

5.5 Cluster Attribution and Distinction

Because we can't previously know what users might search for in their clusters we have to provide them with tools to compare them on their own. To recognise differences between attributes of the found clusters we employed the creation of Box-Plots for every grouped subset of data-points. Box-Plots show the distribution of values for each dimension found inside the subset [WPK89].

While other distribution representations like for example Violin Plots [HN98] might include more information per Plot, we found the simplicity of Box-Plots as a advantageous when comparing multiple Plots with each other.

Box-Plots are Cartesian Plots that contain box elements. These Box-Elements offer a coarse representation of distribution in value sets. All occurring Values of a dimension are sorted in a list from high to low and then the following 5 values define the shape of the Box-Element:

- **max:** The one at the beginning of the list. So the highest one.
- **upper:** The one at the 3/4 th place in the list.
- **median:** The one in the middle, so the median of the represented distribution.
- **lower:** the one at the 1/4 th place in the list.
- **min:** the last element of the list.

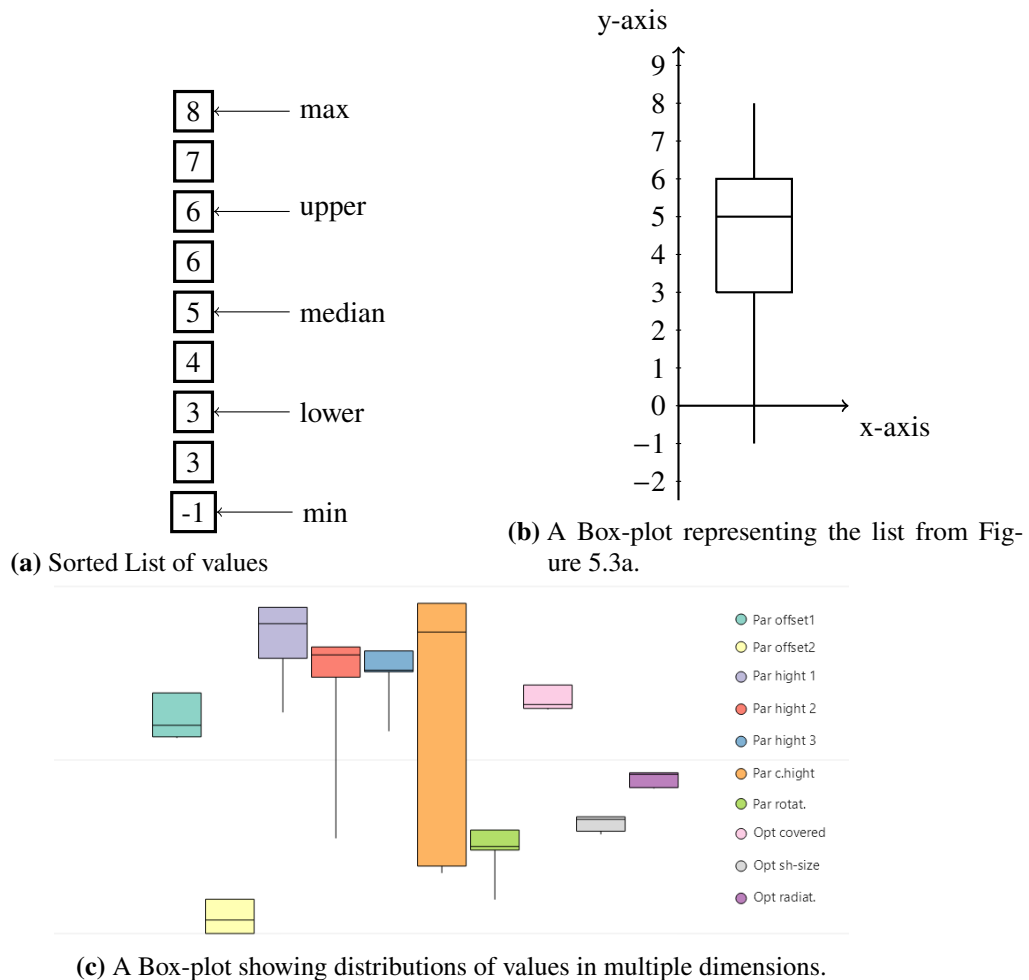


Figure 5.3: Box Plot Explanations: A longer or shorter line on the top or the bottom shows heavy or not-so-heavy outliers. A shorter or longer box means that the bulk (middle 50%) of the values are more or less concentrated. And a the median-line gives us information of the distributions inside of this bulk part.

An explanation for Box-Plots can be found in Figure 5.3.

All Boxplots for computer and human created data groups are displayed in a global list view. The resulting Box-Plot-Matrix allows a quick detection of differences between clusters based on the displayed value distributions as seen in Figure 5.4. Apart from distinction possibilities this global view of all box-plots can also provide knowledge about possible correlations (T3,T4) when only certain combinations of dimensional distributions occur in the generated box plots. A manual creation of new data groups is also part of this view.

For filtration (T1,T2) and further exploration of the data set, the view offers for each data groups:

- to select their members.
- to hide their members.
- to hide all but their members.

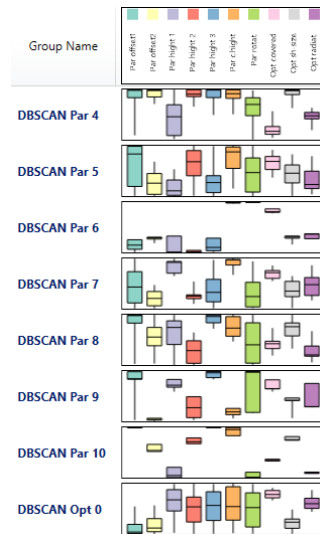


Figure 5.4: A Box-Plot-Matrix showing multiple Box-Plots. It shows distinctive Features of each data group and allows to differentiate and compare them.

The order in which Data-Groups are enlisted is also important if data points are members of multiple Data-Groups. Data-Groups farther down the list can overwrite the Color and Icon preferences of those enlisted earlier. This allows for a flexible workflow with Data-Groups where the visibility of Different Group Memberships can be quickly en- and disabled.

5.6 Bijective Representation

To support the intuitive discovery of data points with known or desirable values (T1,T2) and also the possibility to enter new parameter values (T6) in one view we searched for a bijective presentation. A simple value list could provide this feature and was already in cooperated into opossium. We also opted to include such a view because it offers a intuitive way for showing concrete data point values.

The List view offers two simple list visualizations of the imported data points . In both lists, all Dimensions have their respective Column, and entries can be sorted by them. Sorting by multiple dimensions is also possible for cases where one data point shares the same value in one dimension. Data-Group Memberships of entries are encoded via the row's background color.

All Items List

This list always includes all initially imported data points and allows for the selection of rows. (T2,T1) The selection operates as expected from experiences with popular office applications. Allowing for singular selection, selection of a range of rows, and adding/removing items from an active selection. This list, when the selection capabilities are ignored, offers users a similar experience to the current listing in Opossum.

Selected Items List

This list does not offer selections but instead only lists the currently selected data points. It also shows each dimension's median values for them. Here users can read concrete values of singular or small groups of points.

5.7 Correlation Detection

To allow for correlation (T3,T4) analysis we decided on the inclusion of a Parallel coordinate Plot (PCP) View. [HW13]

Another possibility for the correlation detection would have for example been the usage of a scatter plot matrix(SPLOM) View. [EDF08] Scatter Plot Matrices can disclose the relation of more dimension pairs than PCP at the same time but doing so they also require more screen space. The space requirement of SPLOMs are linear to the amount of displayed dimensions in both spacial directions. The requirement of PCPs on the other hand only grows in one dimension and also far less quickly because an additional axis takes less space than a whole additional row of scatter plots. To circumvent this problem models with more than a certain amount of parameter or objectives would also make user selections of the displayed dimensions necessary in order to keep the plots readable. This introduces startup complexity and we reasoned that by then a more labour intensive but intuitive PCP interaction could even be preferable.

Knowledge about PCPs and their interpretation is also widely spread or could be acquired in a relatively short period. This reduces training periods. Also, their bijectiveness allows for a very intuitive representation of the data points. Users can locate PCP lines of known data points based on their scale values and also derive a general idea of the scale values from marked PCP lines trivially. By changing the order of the Axis in their PCP users can examine possible correlations between dimensions for shown or marked data points. (T3 if Objective Dimensions are shown, T4 if all Dimensions are shown.) Examples can be seen in the Figures 6.2.

By also allowing users to set scale limits in the form of sliders on different axes they are enabled to intuitively select, or filter down to subsets of their data points. (T1,T2) Such a Selection can be observed in Figure 5.5 In Case the Parameter-Dimensions are visible these limits can also be used to restrict the search space for future optimization runs. (T5)

The PCP View right now can only encode one categorization of the data points in their line color. For the second distinction feature of data groups, the Icon Glyphs, we did not find a valid embedding into the PCP view yet. But mouse over pup ups might be a feasible approach in the future.

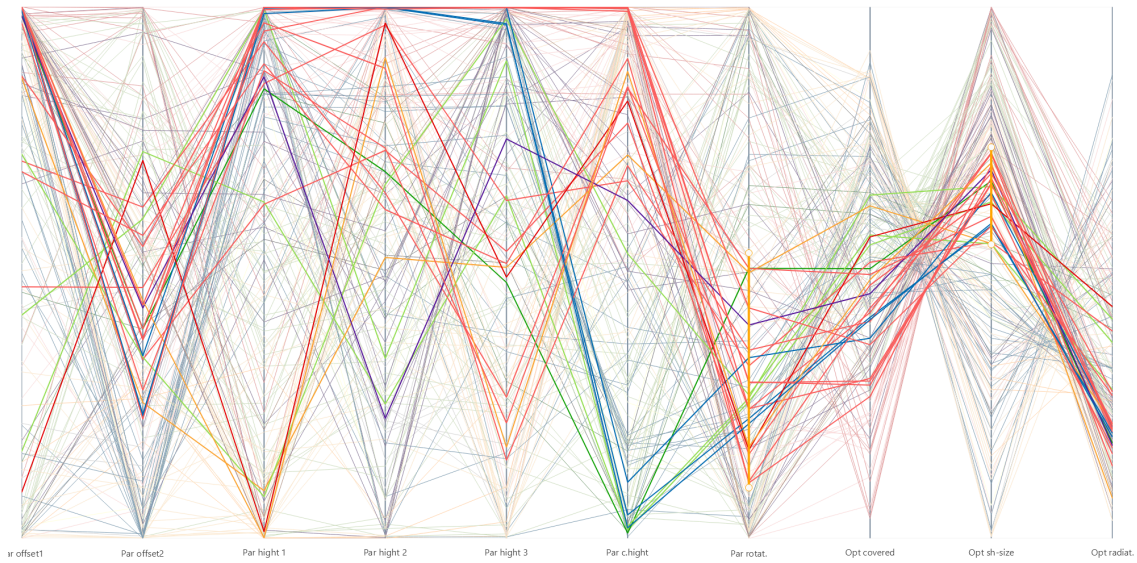


Figure 5.5: Showing a selection done in the PCP-View by limiting certain dimensions. ('Par rotat.' and 'Opt sh-size') The limits can be seen as Orange ranges on the filtered dimensions.

6 Interface Description

In this chapter we quickly explain and go over each view and describe how we envisioned their interaction when working on the given Requirements.

6.1 Modularity

From the interviews we derived that experts like to build their own views if given the chance. We therefore wanted to make all offered View Types as versatile as possible.

Both the Data-Grid-View (1) and the Data-Group View (2) are singular views that are always showing all dimensions, because both of them combined offer the absolute minimum on functionality the program can offer. The PCP-View (3) and the 2D-Cluster-View (4) on the other hand are configurable to only incorporate all dimensions or either the Parameter- or Objective Dimensions.

The modularity would theoretical allow for different Interface Layouts but while implementing and during the evaluation we utilized the configuration found in the screenshot (Figure 6.1).

It contains:

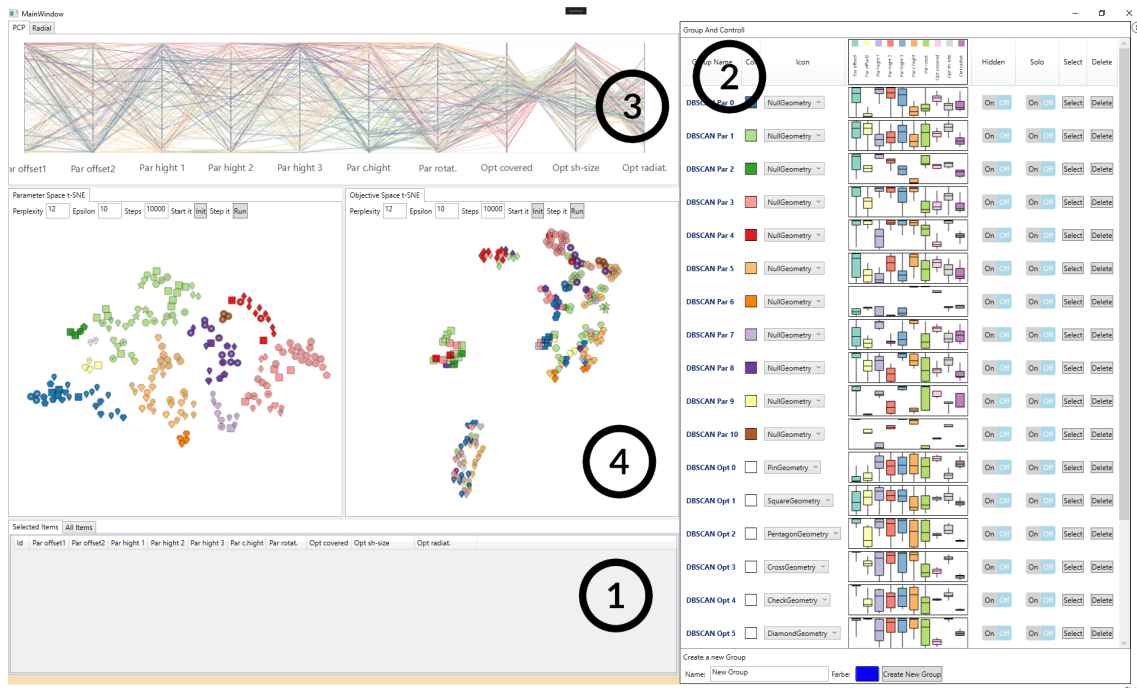


Figure 6.1: A overview of the default user interface.

- a PCP-View (3) that includes all Dimensions of the presented Data.
- two tSNE-Scatterplot-Views (4) next to each other where one includes only the Parameter-Dimensions and the other one only the Objective-Dimensions.

6.2 Use Case

In the interface and the implemented prototype views, we utilize a couple of basic visualization methods that should be described here. Two global concepts are applied for the different Views to work with each other. The application of these Concepts in combination with the Views aims to solve the above-defined Tasks.

Here we disclose how we expect users to work on the defined tasks 4.2 Our starting point is a View-Constellation as seen in the Screenshot in Figure 6.1.

6.2.1 Data-Group View (2)

The Data-Group-View is the place where the actual filtering (T2,T1) of the displayed data can be achieved. New Data-Groups of selected design-options can be created and all Data-Groups allow for the removal of their members from all Views.

Additional usage comes in the form of the Box-Plot Matrix. Analysing them for different Data-Groups can give hints for possible correlations. Box-plots are also not limited to neighbouring Dimensions like the correlations analysis of the PCP can be. This helps with T3 and T4.

6.2.2 Data-Grid-View (1)

It is the only View that offers to read concrete values for data-points.

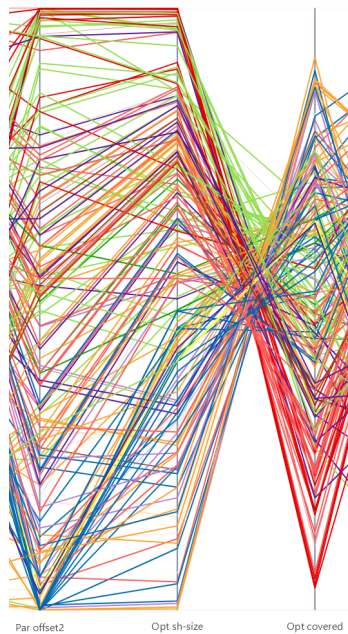
With its sorting and selection capabilities it can be also used to filter out bad performing data-points (T2,T1) by users that prefer this way over the mightier one offered by the PCP View.

In the future it might also be a good place to specify concrete parameter values for a single-point simulation (T6) when the prototype gets integrated into opossium.

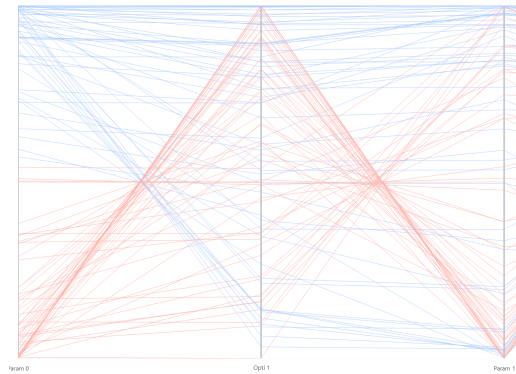
6.2.3 PCP-View (3)

By setting Limiters on one or more dimensions a selection of a subset of PCP-Lines and their corresponding data points is possible. These selections can be grouped into Data-Groups and then either them or all the other data points can be filtered out before further handling. This allows for an iterative filtering of the imported MOO-results like required in T2 and T1.

If the Parameter Dimensions are part of the View setting these Limiters on one or more of the Parameter Axis could also be signaled back to Grasshopper. This would limit the Optimizer for possible future Optimization runs like required in T5.



(a) An example for a simple negative Correlation seen in a PCP.



(b) A more complicated correlation shown in PCP. If 'Param0' is low 'Param1' and 'Opt1' have a negative correlation. But if 'Param1' is high the correlation becomes 'positive'.

Figure 6.2: Two examples for Correlations showing in PCPs.

For Neighbouring Axis PCPs allows for an optical correlation analysis between their visualized Dimensions. This allows for working on T3 if the Objectives are visualized and T4 if both Objectives and Parameters are part of the visualization.

By interactively reordering the axis users are able to analyse correlations between pairs of them. Many possible Correlation Patterns are imaginable to occur here and Architects might have to be trained to properly detect and interpret them. An example for a negative correlation can be seen in Figure 6.2a.

But also by usage of Selection Brushing, by moving the limiters up and down on selective dimensions, such correlations can be detected. PCP lines that cross the filtered axis at the selected height are getting highlighted in the whole view. By moving the selection users are able to detect what effect changes in the filtered dimensions have on the other dimensions. This even allows for the detection of more complex correlations where a correlation between two dimensions might be depending on a third one. Figure 6.2b shows such a complicated correlation.

6.2.4 Splitted 2D-Cluster-View (4)

To gain a detailed level of knowledge over concrete values behind Scatter-Points users have to utilize other parts of the UI. They could look at Box-plots of grouped points in the Data-Group View or they select points in the Scatter plots to inspect the highlighted representations in other views.

The same is true for finding out where certain specific designs ended up after the tSNE Mapping. A selection in other views will highlight the corresponding item in the Scatter plot.

Since the t-SNE method does not allow for a reversal of the mapping a interactive feedback to the optimizer like needed for T5 and T6 are not possible and limits would have to be set in other parts of the interface.

This can also be done the other way around by using the PCP-Selection functionality to highlight points or whole clusters in the 2D-Cluster-Views. By letting tSNE order the Parameter-Value-Sets into a 2D map users gain a 2D representation of look-defining similarities for them to explore. Areas in the resulting mapping could be identified where parameter-vectors define subjectively beautiful, less beautiful designs or even designs with certain visual trades.

In the Objective Space t-SNE tries to arrange similar performing Designs into the 2D representation. Like before users could explore it for areas that are more or less interesting for them based on the balance of their performances.

The automatic clustering offers a quick option for this first step, but the exploration and grouping in both 2D-maps can also be manually achieved by human interaction if desired.

After both categorizations are completed the resulting views can support the user in multiple ways depending on the data. An example could be: If many Elements of a specific Color appear mainly in a small amount of different shapes users can gain a intuition of how designs with similar parameters might perform. This can save them from running cost-intensive simulations.

This concept also works the other way around. If many elements sharing a specific shape are mainly colored in a small amount of colors users could conclude the parameter ranges where to search for buildings that perform a desirable way. Both examples help in narrowing down the simulated parameters in possible future optimization runs(T5).

For the filtration down to a possible final design (T1,T2) this two part categorization is also very useful. Users can filter out bad performing, or aesthetically not pleasing clusters until they are only left with one or a manageable amount of possibilities.

The concrete parameter values of those final designs would have to be read from the Data-Grid-View or the PCP-View.

6.3 Implementation

The Implementation of the Prototype was done in C# utilizing the Windows Presentation Foundation (WPF) Ui-Framework. (<https://github.com/dotnet/wpf>) These requirements were set, given that Oposum is built the same way and we would appreciate a future inclusion into the plugin.

The oposum Plugin and its components are open source which allows the public to change the plugin if they want to. To stay open-source and free to use after including our Vis-Tool made it necessary to select an adequately licensed Graphing library for our Views. After analyzing multiple options we settled on the usage of livechart2. (<https://livecharts.dev/>) Being a modern-looking and very actively developed OpenSource Project itself made working with it easy. In the process of

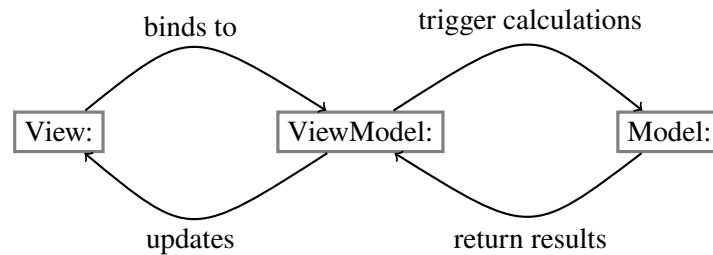


Figure 6.3: Corse Diagram of MVVM-Concept.

our Prototype Development, we had multiple interactions with the livechart2 maintainers. Next to fixing our reported bugs they also accepted pull requests from us and even enabled a use case they previously did not have considered.

We highly advocate for innovators to extend and adjust our Multi-View Platform to their liking. To support future development proceedings we tried to strictly incorporate the Model-View-ViewModel (MVVM) coding style.

6.3.1 MVVM

Model-View-ViewModel (MVVM) is a concept known to WPF where Code-Parts that describe different aspects of an interactive UI are strictly separated. (<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>) The 'View' is the actual visual Part, so the arrangement, size, or color of different UI components. Different parts of the 'View' like Numbers, simple List Entries, and also Interaction-Events are bound to Properties and Actions of a 'ViewModel'. Here preprocessing and interaction with data are handled.

The actual data and algorithms are described in the 'Model'. The Data Handling is triggered by the 'ViewModel'. A Corse Diagram explaining the Concept can be found in Figure 6.3.

For us, the model part includes the formal MOO results, the Data-Groups, and the current selection. The View of the 2D-Cluster-View defines the look of the Cartesian Scatter Plot. Apart from this it binds the Scatter-Plot-Elements to a List Property and defines which Actions to call should certain Mouse interaction occur. The whole Dimension-Reduction and Clustering Steps of data points as well as the interaction handling is done in the ViewModel.

Instead of designing whole new View + ViewModel bundles. Possible contributors are encouraged to simply derive a new ViewModel for the 2D-Cluster-View that implements a different Dimension-Reduction Methode or a different Clustering Algorithm. As long as it offers the necessary Properties and Actions the existing View can be bound to their new View-Model saving them the design work.

6.3.2 Data Structure

To allow for even easier implementations in the future we tried to structure the code in a certain way to make working with the created Structure easier. For example every View Model class inherits its special properties and functionalities from interfaces and base classes. This allows for similar Interface Modules to share big parts of their code. We hope that Future development tasks turn out to be easier to implement because of this prepared framework.

7 Evaluation

To gain feedback and as evaluation for our prototype we organized a hands on review of the tool with domain expert that has multiple years of experiences in architecture field and is familiar with opossum. This chapter contains a description of the procedure from this event as well as the given feedback to different parts of the visualizations.

7.1 Procedure and Tasks

The review process started with a quick explanation of the tool, its views and the interaction capabilities. Expansions of the utilized technologies (PCP and tSNE) where not necessary since a good understanding of them were stated by the reviewer in a short questionnaire beforehand. Afterwards the field expert tried to solve the collected requirements via the prototype.

We asked the experts to perform three tasks:

- Find Correlations (T3 and T4)
- Filter the optimization results to ones that would be interesting to you. (T1)
- Find possible Parameter-Limitations for future optimization runs. (T2 and T5)

Since, at the stage of the evaluation an integration of the prototype into Opossum has not taken place the review was done on two predefined data-sets of MOO result exports. On both data sets the field-expert tried to solve the currently possible subset of the previously mentioned tasks. For the missing tasks (T5 and T6) we asked them to verbally mention what they would have configured.

After and during the hands-on testing of the tool the domain expert was encouraged to give feedback in verbal form. The spoken audio and the screen-contents of the 90 minutes session where recorded for later evaluation. After the provided feedback was summarized into this theseis the recordings were deleted.

7.2 Datasets

The two data sets contained MOO-result exports originating from two different Grasshopper Models.

1. a simple model with pre-known correlations.
2. a more complex model provided by the domain experts as a realistic example.

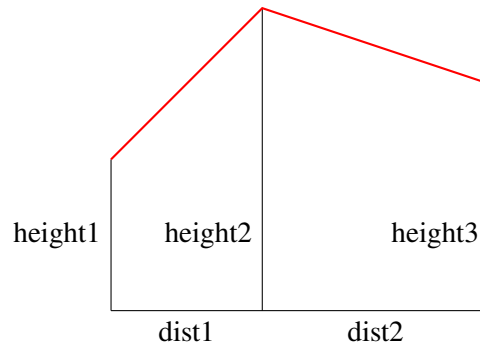


Figure 7.1: Simple Evaluation Model

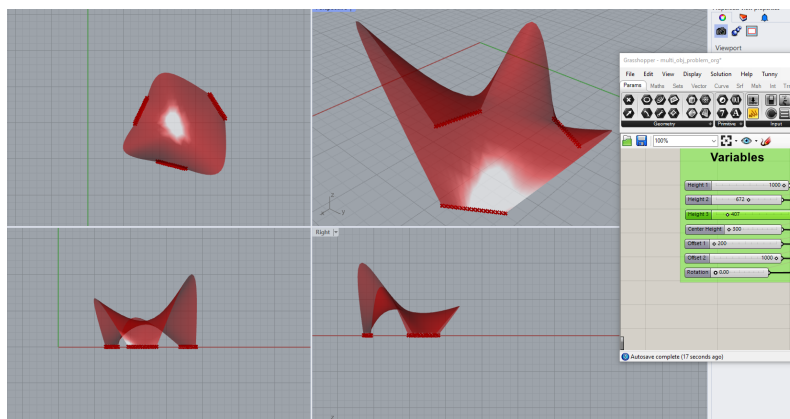


Figure 7.2: Realistic Evaluation Model opened in Rhino/Grasshopper.

7.2.1 Simple Model

For a easy comprehensibility we hand crafted a simple Model describing a 2D Figure. The Model can be seen in Figure 7.1.

The parameters and their ranges are:

- height{ 1,2,3} [0.0 - 1.0]
- distance{ 1,2} [0.2-1.0]

The optimization goals are:

- maximize Aesthetic = length(red_line) + ((height1 + height3)/2) as an arbitrary search for a specific trade. This prefers a high up but also also sloping (to get longer) roof.
- maximize the area the figure takes.
- maximize Cost = $-1 * (\text{height1} + \text{height2} + \text{height3})$ as a function for cost. Because opossum can only maximize or minimize we had to maximize the negative value.

We opted for these optimization goals to be able to detect different correlations between different pairs of dimensions. For example will the cost function dictate for a smaller area and the other way around forming a directly negative correlation between them.

7.2.2 Realistic Model

The more realistic Model was provided by the field experts for one of the iterative design meetings. It describes a Canopy, that can be described as a hull with 3 arch-formed entrances or holes. The structure gets described by 7 parameter:

- height{1,2,3} [300 - 1000] describing the heights of arches.
- center height [300 - 1000] sets the height of the canopy center.
- offset1 [200 - 500] plays a factor in the distance between the holes at ground level.
- offset2 [100 - 1000] together with offset1 plays a factor in how far the arches are leaning into the center or away from it.
- rotation [0.0 - 360.0] rotates the model around its center.

The optimization results we received had the following three objectives that all were tried to minimize by the Optimizer.

- Covered: The Area the Canopy covers. Multiplied by -1 to maximize instead of minimize.
- Shellsize: The Size of the Canopy. Minimize to reduce building costs.
- Radiation: Describe the simulated amount of Solar Radiation that hits the area. A minimization here searches for Canopys that are effective in blocking the sun.

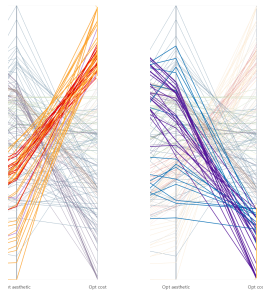
More than these information regarding their calculation were not known by us. Whenever the dataset was utilized for visual reference thru-out the development phases, it's dimensions were presented without their names (Para 1, Para 2 ... Opti 1 ...). This allowed us to be neutral when talking to the field expert because we also did not know about possible properties or correlations of the data-set. For working with the data-set we also provided the domain expert with the loaded Model in Rhino so he could understand the 3D Model and its Parameters.

7.3 Findings

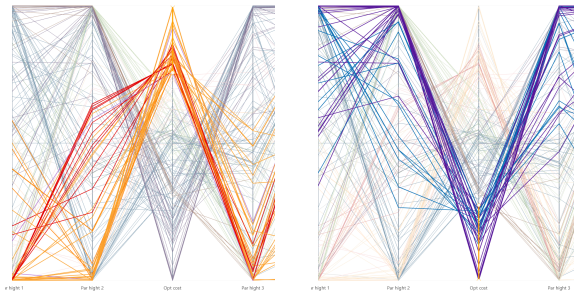
In the following we summarize the analysis findings for each task. We start with the simple model followed by the realistic one.

7.3.1 Simple Model

To get better insights into the Correlations between the Dimensions instead of moving Axis next to each other he analyzed the effects of Selection Brushing in the PCP View. By doing so he found the direct negative correlation between 'Cost' and 'Area' (T3). After asking him to find correlations between Parameters and Objectives he searched with selection brushing and axis moving he found the negative correlation between 'height1', 'height2' and 'height3' and 'Cost'. (As a reminder the 'Cost' values are negative, see Section 7.2.1.) Recreations of the observed interactions can be seen in Figure 7.3a and 7.3b.



(a) Brushing utilized for the detection of a negative correlation between Objectives (T3).



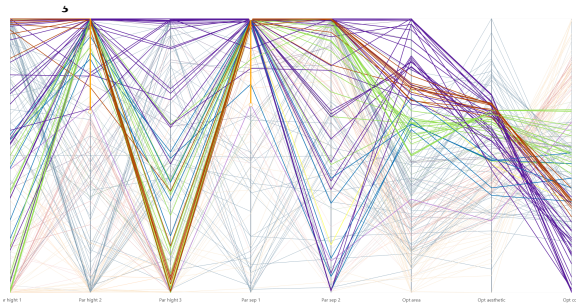
(b) Brushing utilized for the detection of negative correlations between one objective and three parameters. (T4)



(c) Screenshots of the interface with limited PCP axis and after the pentagon-Geometry cluster was detected and selected.



(d) A screenshot for the tool after the Gem-Geometry cluster was selected.



(e) A recreated PCP View with the configured limiters for Task T5.

Figure 7.3: Recreations of the interaction steps utilized by the domain expert on the simple models MOO results during the evaluation.

For the second task of finding and filtering down the results to those he would find interesting based on their performance he also used the limiter functionality in PCP. By then he would have solved Task T1 already but he also noticed that nearly all of the highlighted Elements in the Objective-2D-Cluster-View shared the same Icon-Symbol: *“I see now I could have also just select it here. That would have been faster”* (T1). He then changed the selection to the Pentagon-Geometry Cluster which was the one with relatively high performance values in all three Objectives. A screenshot of interface before and after the selecting the pentagon-geometry cluster can be found in Figure 7.3c.

For the third task he should limit the parameters for a future optimization run in search for high 'Area'. By comparing Box-Plots in the Box-Plot-Matrix he directly selected the Objective Cluster with the high values in 'Area' (T1). This can be seen in Figure 7.3d, t *“Nearly all of them are purple so they are from the same Parameter Cluster as well.”* He analyzed the highlighted Lines in the PCP and then configured limiters for high 'height2' and high 'distance1' values there (T2). After analyzing the resulting selection he concluded that these limits would be his choices to shrink the search space (T5). His selection was recreated in Figure 7.3e. With our knowledge over the model, we can evaluate this choice and know that both limitations definitively are advantageous for the 'Area'.

Interestingly 'distance1' and 'distance2' by design both share the same correlation to 'Area' and also are positively affecting 'Aesthetic'. They both also do not influence the 'Cost' performance. But while the optimizer clearly found the advantage and simulated a lot of designs for high 'distance1' values it seemed to have had problems in detecting a clear preference for specific 'distance2' values. This can also be observed in Figure 7.3e.

7.3.2 Realistic Model

Directly after the realistic model loaded the expert mentioned *“This is very good to know for MOO because they are clearly negative correlating”*(T3) when he saw a lot of crossed lines between the objectives 'covered' and 'shell size', (See Figure 7.4a). The second part of the task, finding correlating Parameters for the objective 'shell size', (T4) was solved by selection brushing on this dimension first to get possible candidates and then by moving the objectives axis in the Parallel Coordinate Plot to analyze the suspected relationships. His first conclusions were: *“For 'offset2' we can see a positive influence, because the lines are mostly horizontal. [...] for 'offset1' its not that clear”*. But after filtering (T2) to first high and then low 'offset1' values: *“There is a correlation but not as strong as for 'offset2'”*, (See Figure 7.4d).

He then started his search for Models with a good balance in their performances (T1) by analyzing the Box-Plot-Matrix. After selecting a interesting cluster and further PCP filtering he noticed a lot of crossed lines between 'offset1' and 'offset2'. He analyzed this fact via axis moving and via selection brushing (T2) and also tried it out in the Rhino-Model, see (Figure 7.4b and Figure 7.4e). He then concluded that for data points to perform in this preferable way they need to have: *“Either a high offset1 and a low offset2 value or the other way around”*. He also mentioned: *“this is very interesting, that we actually found such a complicated relationship”*(T4).

When trying to deduct a possible narrowing of the parameter space (T5) he analyzed the Box-Plot matrix and noticed that the distribution for 'rotation' in nearly all Performance Clusters spanned the whole possible value room, (See Figure 7.4c). He then tested it in the Rhino model and found out that the 'rotation' parameter was not part of any objective calculation. Knowing MOO Problematic he said: *“This is bad, the optimizer will have a much harder time to find good models that way. But good that we can find indications here.”*

After this deciding on a performance cluster (See Figure 7.4f) and after brushing in the PCP concluded with the limiting of 'offset2' to low and 'offset1' and 'height1' to high values, (See Figure 7.4g).

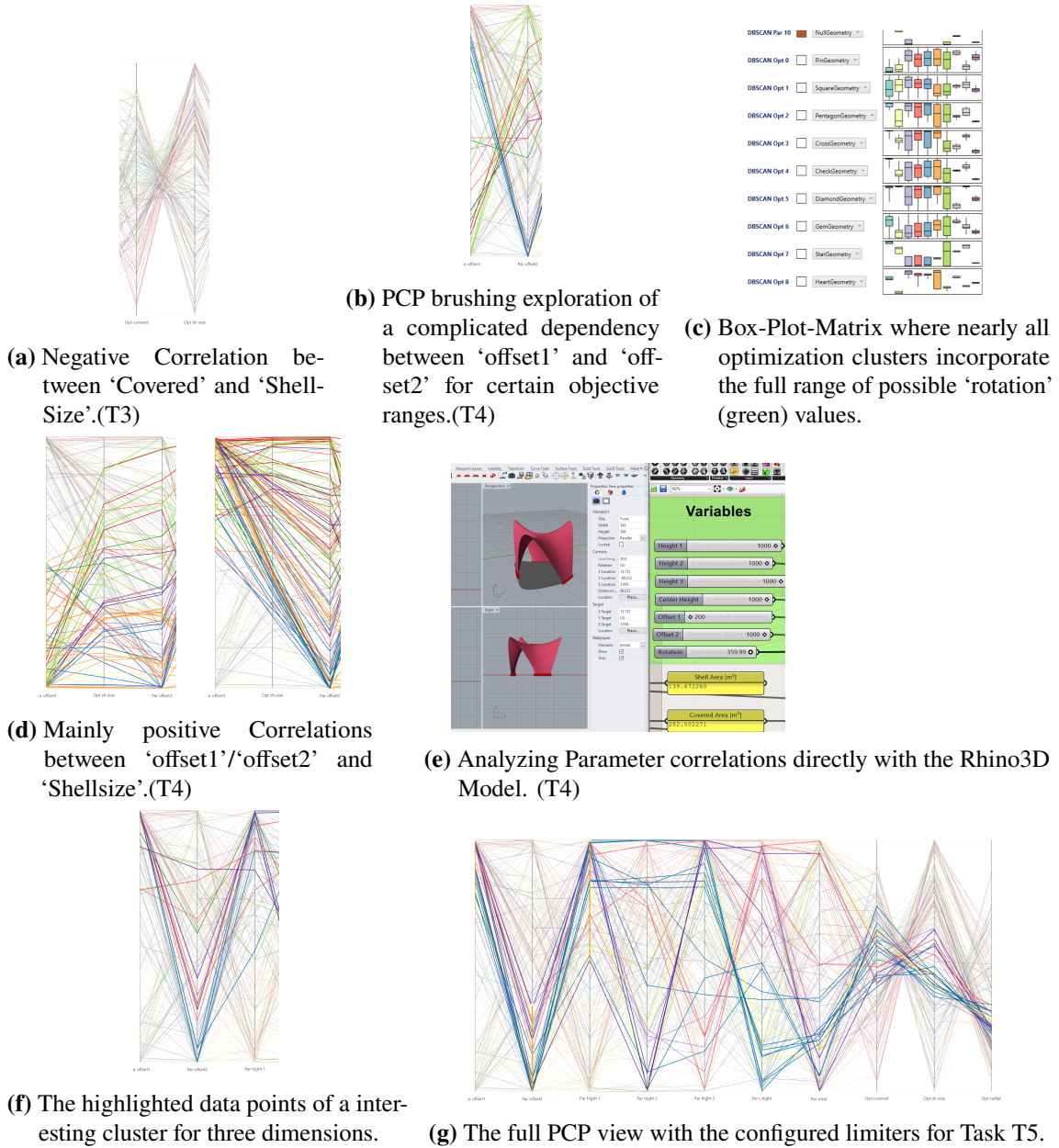


Figure 7.4: Recreations of the interaction steps utilized by the domain expert on the realistic models MOO results during the evaluation.

8 Discussion

In this chapter, we discuss the received feedback from the domain expert and our own findings. We will also comment on the open problems and issues of the prototype and how we plan to address them in the future.

8.1 Expert Feedback

Throughout the multiple development meetings and also in the evaluation we collected feedback on our tool that we want to discuss.

A field expert in the development interviews summarized the functionality with an overall positive statement: *“The tool overall does solve the collected problems we see with opossum right now”*. Also, the participant in the expert review successfully solved all tasks that were issued to him and expressed his appreciation during certain parts of the procedure. (see Chapter 7)

But next to the overall positive feedback were also mentions of open points that need to be addressed. Some have to be solved before a release and some that can also be addressed in later versions. We tried to categorize them into two Sections:

- Necessary Improvements before a possible release
- Smaller Feature requests

8.1.1 Necessary Improvements

Most of the given feedback was related to the missing direct link between Rhino3D or Opossum and our Visualization, (See Section 8.2). Without it, Task T5 and Task T6 are not functional and only work on a theoretical level by manually configuring the Grasshopper Model.

An additional point mentioned in both the evaluation and the review meetings is the inclusion of a 3D Rendering of selected parameter value sets. The practice to modify the grasshopper model by hand to look at a design is not practicable. Such rendered models can be generated via Rhino3D like other existing plugins already demonstrate but also requires a Rhino3D integration. This would hugely benefit the evaluation of described designs, and therefor exploration of the parameter space visualizations. Both single point renderings as well as renderings of average parameter values for selections or data groups are imaginable.

8.1.2 Feature Requests

The feedback issued for specific views originated mainly from the expert review.

While working with the 2D-Cluster-View (section 6.2.4) the expert mentioned: *“I miss a zoom functionality, because it is hard to draw the selection polygon without including [unwanted objects]”*. He also miss a possibility to snap all user moved scatter points back to their original clustered place. Both functionalities should be possible to integrate and are planned for future versions.

For the PCP view the evaluation showed that the nature of the objective part of the Optimization results can be difficult to understand. Because Opossum in it’s current form can only aim to either maximize or minimize all given objectives some simulation results had to be negated (multiplied with -1) before. This has the effect that Optimization Dimensions like for example ‘Cost’ are not intuitively understandable in their PCP representation. The comment of the reviewing expert: *“I want to reduce costs I assume, but is up good or down?”* summarizes the Problem pretty good. Because we can’t relay change the way opossum functions a possible option also discussed with the review expert would be to let users flip PCP Axis on will.

When interacting with the Data Group View the expert mentioned: *“It would be nice to select two Data-Groups and only get the points contained in both.”*. This functionality would have also speed up some of the observed filtering (T1 and T2) actions from the review. The alternative after some trial and error was a multi step approach:

- solo out the first data group. (Hides all other points)
- select the second data group.
- create a new combined group.
- disable the soloing to show all points again.

He also said: *“After selecting a performance cluster i would be interested in what would happen to the objective Values if I choose a different value for a parameter.”* We understand this as as a specialized case of Correlation detection (T4). With the currently offered Views we could imagine to provide an option to derive PCP parameter dimension filters from data groups. Users could then move these filters up and down to observe the effect this would have on the objectives. Or they could then remove only the filter for the parameter they are interested in. By doing so they would broaden their selection and could then search for Correlations by moving the objective axis next to it. On a local level (relatively small parameter changes) the described BIOT technique (see Section 3.2.2) could also be of help here.

8.1.3 Limitations

Because of time constraints we carried out a single person expert review. We are aware that this kind of review alone can only be given little attention when estimating the prototypes contribution.

But even if regarded under these limitations we hope that this documented hands on trial was able to proof that the tools capabilities are helping architects to interpret their MOO results.

After an Integration into Opossum and after the points in Section 8.1.1 are resolved we would highly encourage a wider spread user study with multiple architects to collect quantifiable evaluation data.

8.2 Grasshopper Integration

In the beginning of the Development Phase we decided to develop our prototype as a standalone Program first. We did so to speed up the startup process for every debug run. Before the split every startup included:

- The compilation time of the combined WPF Application (see Section 6.3).
- The startup time of Rhino3D and Grasshopper.
- The loading time for a MOO Model.
- The optimization run for Opossum.

Because of these reasons we started to work for a prototype that imports the MOO results rather than directly receiving them from the opossum grasshopper element.

By the time we had to close development for the interface a reintegration would have taken to much time. So while the above arguments are still valid a earlier reintegration into opossum would have allowed us to implement the back communication part of Task T5 and Task T6.

Theoretical practicability of these concepts is proven by similar functionalities being present in development branches of the opossum code repository.

8.3 Future Work

We disclosed part of our decision making already in Section 5.

But some improvements we could see in the future are:

- **Hierarchical Clustering:** with a clever interaction concept could offer better clustering than the now used DB-Scan algorithm.
- **SPLOM:** The inclusion of SPLOM (Scatter Plot Matrices) as an additional View for easier Correlation Detection. But as described in the Design Space a solution for Selecting Dimensions would be necessary.
- **Handling of Dimensions:** we mapped all dimensions (qualitative and quantitative) into quantitative dimensions in \mathbb{R} , a better mapping, maybe user configured might be helpful here.
- **Scatterplot Encodings:** Right now we encode two cluster memberships in our 2D-cluster Views using different Colors and Icons. But it could also be preferable to use the Icon Property to show a value dependant Glyph that allows users to extract direct information about dimensional values without the need of selecting the data point first. A possibility would be to let the user decide how and what property they want to define their encoding.

- **BIOT:** Described before we think that the BIOT technique can be utilized to help understand the non linear mappings seen in the 2D-cluster-views. But before this can be applied as a general feature for this tool we have to find answers to occurred questions. Most importantly about how we could incorporate the information provided by the BIOT calculations without cluttering the 2D-View and without overwhelming/confusing the users.
- **Performance Map:** The estimation capabilities for single objectives is a feature we do not provide until now. But it would be imaginable to include ideas of Performance map into our work in the future. Multiple Performance Maps next to each other would provide architects with additional hints where to search with the next simulation run. (T6)

We are aware that different designers might prefer different techniques and also that the use case for the tool could undergo changes in the future that require different tools. Because we aim to open source our code together with opossum we planned for an easy adaptability from the start. (see Section 6.3) With the deployed class structure we tried to create an easy to adapt MOO analysis framework. Every change of technique but also every additional view should be realizable without reading or understanding 100% of the source code. By lowering these, sometimes daunting, requirements we hope to encourage future improvements and developments.

9 Conclusion

Throughout the interviews with the domain experts, we were able to get a deeper understanding of the shortcomings of the current MOO Process in architectural design. Together with them, we derived a list of requirements and developed an interactive visualization prototype.

The prototype proved its capability of helping with or solving these requirements in a carried-out expert review. However, more intense user studies should be deployed before final conclusions about its quality can be drawn.

Given the chance by the future planned open sourcing of the opossum code base we also planned ahead. To prepare for future requirement changes and for the adaption of additional visualisation approaches we constructed a class and interface structure that can be utilized as a framework for us or future developers. Strictly believing in the philosophy behind open source we also hope and plan to continuously work together with the architectural community in improving this tool and opossum in general.

We hope that after we take the remaining steps and finish the integration into opossum our visualization becomes part of future architectural building design pipelines. And that it provides the rhino users with the crucial missing link between the human architect and their computer coworker in their collective task of finding a multi-objective optimized building Design.

Bibliography

- [AR22] L. Anand, C. M. Rodriguez Lopez. “ChromoMap: an R package for interactive visualization of multi-omics data and annotation of chromosomes”. In: *BMC bioinformatics* 23.1 (2022), p. 33 (cit. on p. 11).
- [BC87] R. A. Becker, W. S. Cleveland. “Brushing scatterplots”. In: *Technometrics* 29.2 (1987), pp. 127–142 (cit. on p. 22).
- [BMSF21] A. Bibal, R. Marion, R. von Sachs, B. Frénay. “BIOT: Explaining multidimensional nonlinear MDS embeddings using the Best Interpretable Orthogonal Transformation”. In: *Neurocomputing* 453 (2021), pp. 109–118 (cit. on p. 17).
- [CAB+23] R. Crassard, W. Abu-Azizeh, O. Barge, J.É. Brochier, F. Preusser, H. Seba, A. E. Kiouche, E. Régagnon, J. A. Sánchez Priego, T. Almalki, et al. “The oldest plans to scale of humanmade mega-structures”. In: *Plos one* 18.5 (2023), e0277927 (cit. on p. 9).
- [EDF08] N. Elmqvist, P. Dragicevic, J.-D. Fekete. “Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation”. In: *IEEE transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1539–1148 (cit. on pp. 15, 28).
- [HN98] J. L. Hintze, R. D. Nelson. “Violin plots: a box plot-density trace synergism”. In: *The American Statistician* 52.2 (1998), pp. 181–184 (cit. on p. 25).
- [HW13] J. Heinrich, D. Weiskopf. “State of the Art of Parallel Coordinates.” In: *Eurographics (State of the Art Reports)* (2013), pp. 95–116. doi: [10.2312/conf/EG2013/stars/095-116](https://doi.org/10.2312/conf/EG2013/stars/095-116) (cit. on pp. 15, 22, 28).
- [HZS+23] H. Heberle, L. Zhao, S. Schmidt, T. Wolf, J. Heinrich. “XSMILES: interactive visualization for molecules, SMILES and XAI attribution scores”. In: *Journal of Cheminformatics* 15.1 (2023), pp. 1–12 (cit. on p. 11).
- [JC16] I. T. Jolliffe, J. Cadima. “Principal component analysis: a review and recent developments”. In: *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202 (cit. on p. 23).
- [Kan00] E. Kandogan. “Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions”. In: *Proceedings of the IEEE information visualization symposium*. Vol. 650. Citeseer. 2000, p. 22 (cit. on p. 15).
- [KW78] J. B. Kruskal, M. Wish. *Multidimensional scaling*. 11. Sage, 1978 (cit. on p. 23).
- [Rob07] J. C. Roberts. “State of the art: Coordinated & multiple views in exploratory visualization”. In: *Fifth international conference on coordinated and multiple views in exploratory visualization (CMV 2007)*. IEEE. 2007, pp. 61–71 (cit. on p. 22).
- [Sch07] M. A. Schnabel. “Parametric designing in architecture”. In: *Computer-aided architectural design futures (CAADFutures) 2007: proceedings of the 12th international CAADFutures conference*. Springer. 2007, pp. 237–250 (cit. on p. 13).

- [SMM12] M. Sedlmair, M. Meyer, T. Munzner. “Design study methodology: Reflections from the trenches and the stacks”. In: *IEEE transactions on visualization and computer graphics* 18.12 (2012), pp. 2431–2440 (cit. on p. 19).
- [TD+10] S. Tornincasa, F. Di Monaco, et al. “The future and the evolution of CAD”. In: *Proceedings of the 14th international research/expert conference: trends in the development of machinery and associated technology*. Vol. 1. 1. Citeseer. 2010, pp. 11–18 (cit. on p. 9).
- [WCW22] T. Wortmann, J. Cichocka, C. Waibel. “Simulation-based optimization in architecture and building engineering—Results from an international user survey in practice and research”. In: *Energy and Buildings* 259 (2022), p. 111863 (cit. on p. 9).
- [WF20] T. Wortmann, T. Fischer. “Does architectural design optimization require multiple objectives? A critical analysis”. In: *RE: Anthropocene, Design in the Age of Humans- Proceedings of the 25th CAADRIA Conference* 1 (2020), pp. 365–374 (cit. on pp. 10, 13).
- [Wol19] J. Wolfartsberger. “Analyzing the potential of Virtual Reality for engineering design review”. In: *Automation in Construction* 104 (2019), pp. 27–37 (cit. on p. 9).
- [Wor17a] T. Wortmann. “Opossum - Introducing and Evaluating a Model-based Optimization Tool for Grasshopper”. In: *Protocols, Flows, and Glitches - Proceedings of the 22nd CAADRIA Conference* (Hong Kong, CN). Ed. by P. Janssen, P. Loh, A. Raonic, M. A. Schnabel. CAADRIA, 2017, pp. 283–292 (cit. on pp. 9, 14).
- [Wor17b] T. Wortmann. “Surveying design spaces with performance maps: A multivariate visualization method for parametric design and architectural design optimization”. In: *International Journal of Architectural Computing* 15.1 (2017), pp. 38–53. DOI: [10.1177/1478077117691600](https://doi.org/10.1177/1478077117691600) (cit. on pp. 10, 14, 18).
- [WPK89] D. F. Williamson, R. A. Parker, J. S. Kendrick. “The box plot: a simple visual method to interpret data”. In: *Annals of internal medicine* 110.11 (1989), pp. 916–921 (cit. on p. 25).
- [WVJ16] M. Wattenberg, F. Viégas, I. Johnson. “How to Use t-SNE Effectively”. In: *Distill* (2016). DOI: [10.23915/distill.00002](https://doi.org/10.23915/distill.00002). URL: <http://distill.pub/2016/misread-tsne> (cit. on p. 16).

All links were last followed on October 30, 2023.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature