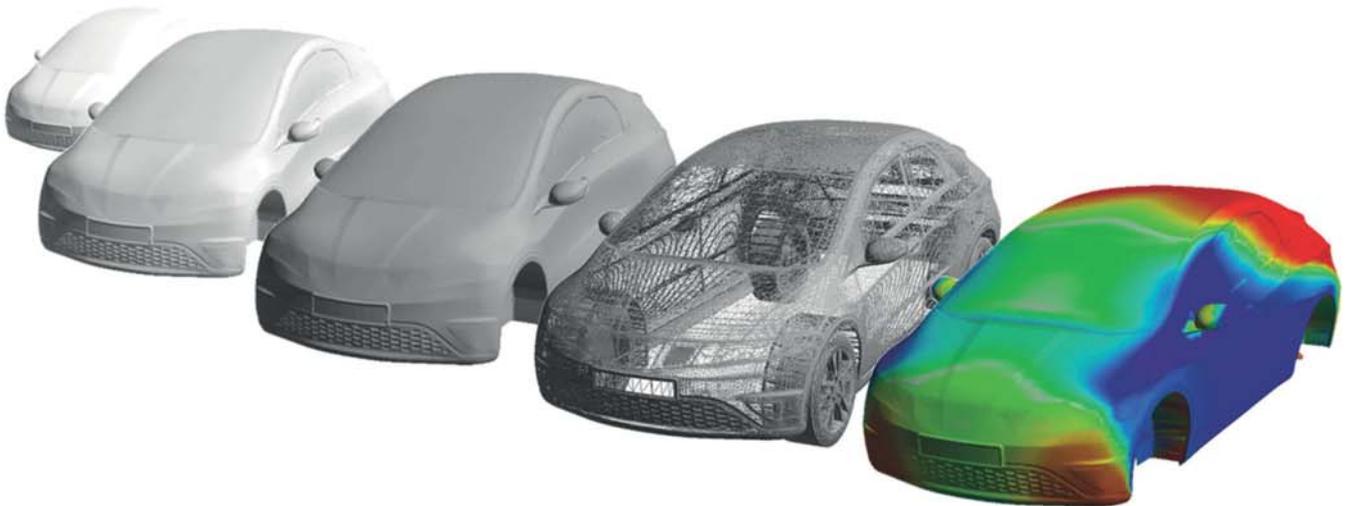


Lars Gräning

# *Shape Mining*

Knowledge Extraction from  
Engineering Design Data



Cuvillier Verlag Göttingen  
Internationaler wissenschaftlicher Fachverlag



# Shape Mining





# Shape Mining

Knowledge Extraction from Engineering Design Data

Lars Gräning

Dissertation submitted to the  
Faculty of Technology  
Bielefeld University  
in partial fulfillment of the requirements for the degree of

*Doktor der Ingenieurwissenschaften*

*(Dr.-Ing.)*

May, 2014



## **Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

1. Aufl. - Göttingen: Cuvillier, 2014

Zugl.: Bielefeld, Univ., Diss., 2014

© CUVILLIER VERLAG, Göttingen 2014

Nonnenstieg 8, 37075 Göttingen

Telefon: 0551-54724-0

Telefax: 0551-54724-21

[www.cuvillier.de](http://www.cuvillier.de)

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage, 2014

Gedruckt auf umweltfreundlichem, säurefreiem Papier aus nachhaltiger Forstwirtschaft

ISBN 978-3-95404-817-5

eISBN 978-3-7369-4817-4



## Acknowledgements

Looking back, I never would have seen myself writing the acknowledgements of my successfully defended thesis, and this definitely would not have happened without the support of a great number of people. When I started to write down all the names, I began to feel grateful and was deeply touched.

First and foremost, I would like to express my deep gratitude to Prof. Dr. Bernhard Sendhoff, president of Honda Research Institute Europe (HRI-EU). You have always encouraged me to keep on writing and to finally submit my dissertation. Thank you for being supervisor, scientific adviser and motivator at the same time. Even when you had tons of work to do, you managed to reserve some of your valuable time for me.

I would like to thank the examination committee, especially Prof. Dr. Helge Ritter, who gave me the chance to submit my dissertation to the university of Bielefeld. After each discussion, I left Bielefeld with a different perspective on my research questions. I am very grateful to Prof. Dr. Yaochu Jin, who prompted and influenced my whole research career. Thank you for your guidance. Additionally, I would like to thank Prof. Dr. Mario Botsch for leading the examination committee and for patiently answering my emails with questions related to computational geometry. I also like to send out a big thanks to Dr. Benjamin Inden, who accepted to join the examination committee.

I am obliged to Prof. Dr. Edgar Körner and Andreas Richter, without whom I would have not started my work at HRI-EU. You gave me the opportunity to be working in a delightful research environment.

I would like to thank all my colleagues, who supported me in finalizing my thesis. Many of them became really good friends. Specially, I would like to thank Dr. Markus Olhofer, who supported me right from the beginning, and Dr. Stefan Menzel, who helped me a lot, such as setting up the application scenario. Many valuable ideas from my thesis are the result of our discussions. A special thanks goes to Kaname Narukawa and his lovely wife Emiko. Thank you for teaching me about Japanese culture and for all the time we spent together.

Furthermore, I would like to express my gratitude to Dr. Sebastian Schmitt, Dr. Nils Einecke, Dr. Tobias Rodemann, Dr. Heiko Wersing, Olga Smalikhov, Nikola Aulig, Nevriye Memet, Andrea Voit, Burkhard Zittel, Dr. David Luttrupp and Elshah Hrnjic, who all aided me in many different ways. It is in my heart to thank also my Japanese colleagues, especially, Dr. Toshiyuki Arima and Dr. Takahiro Fukushige, as well as my former co-workers Dr. Till Steiner, Dr. Giles Endicott, Dr. Lisa Schramm and Dr. Ingo Pänke.



---

Above all, and it is one of my deepest concerns, I like to thank my darling Evy. Thank you for all your patience, your mental support, our discussions at the balcony and for all the love you give to me. You are the most lovely girl in the world. This thesis is dedicated to you.

Thanks, to my parents Uta and Peter, not only for giving me birth but also for allowing me to find my own way. I like to thank my brother Sven and his wife Jessi, my niece Leo, Sigrid and Detlef, my parents-in-law, as well as Sebastian. Sebastian, you deserved more titles or prizes than I do for fighting back to life. Compared to your achievements writing a thesis is a relative doddle.



	Page
<b>1 Introduction</b>	<b>5</b>
1.1 Motivation . . . . .	5
1.2 Overview . . . . .	6
<b>2 Design Synthesis</b>	<b>9</b>
2.1 Engineering Design . . . . .	9
2.2 Computer Aided Design Synthesis . . . . .	13
2.3 Data Analysis . . . . .	14
2.3.1 Data Mining . . . . .	15
2.3.2 Data Mining for Aerodynamic and Structural Design . . . . .	17
2.4 Passenger Car Design Synthesis . . . . .	18
2.4.1 Design Representation . . . . .	18
2.4.2 Computational Fluid Dynamics Simulation . . . . .	22
2.4.3 Explorative Design with Latin Hyper Cube Sampling . . . . .	24
2.4.4 Exploitive Design with an Evolution Strategy . . . . .	25
2.5 Interim Summary . . . . .	30
<b>3 Meta Design Representation</b>	<b>31</b>
3.1 Unstructured Surface Mesh . . . . .	31
3.2 Local Surface Differences . . . . .	33
3.2.1 Vertex Displacement . . . . .	34
3.2.2 Curvature Disparity . . . . .	36
3.2.3 Identification of Corresponding Vertices . . . . .	38
3.3 Data Generation . . . . .	42
3.4 Quantification and Analysis of Design Variations . . . . .	45
3.4.1 Relative Mean Vertex Disparity . . . . .	45
3.4.2 Relative Design Variance . . . . .	46
3.5 Application to Passenger Car Design . . . . .	47
3.5.1 Approaching Design Similarities . . . . .	48
3.5.2 Identifying Weak Deformed Design Spots . . . . .	49
3.5.3 Evaluating the Course of Design . . . . .	52
3.6 Interim Summary . . . . .	54
<b>4 Design Sensitivity Analysis</b>	<b>57</b>
4.1 Sensitivity Estimation . . . . .	57
4.1.1 Linear Correlation Techniques . . . . .	59
4.1.2 Mutual Information Based Sensitivity Index . . . . .	60
4.1.3 Robust Estimation of Mutual Information . . . . .	63



# CONTENTS

---

4.2	Identification of Sensitive Areas . . . . .	66
4.2.1	Unsupervised Clustering . . . . .	67
4.2.2	Measure of Vertex Similarity . . . . .	69
4.2.3	Automatic Identification of Sensitive Areas . . . . .	71
4.2.4	Example . . . . .	72
4.3	Utilizing Sensitivity Information . . . . .	72
4.4	Application to Passenger Car Design . . . . .	73
4.4.1	Direction of performance improvement . . . . .	74
4.4.2	KNN Sensitivities . . . . .	77
4.4.3	Reliability of Probabilistic Sensitivity Estimates . . . . .	80
4.4.4	Sensitive Areas . . . . .	82
4.5	Interim Summary . . . . .	84
<b>5</b>	<b>Design Concept Retrieval</b>	<b>85</b>
5.1	Definition . . . . .	85
5.2	Concept Representation . . . . .	87
5.3	Concept Evaluation . . . . .	88
5.3.1	Relevance and Interestingness . . . . .	89
5.3.2	Measure of Utility . . . . .	90
5.4	Concept Retrieval . . . . .	94
5.4.1	Decision Tree . . . . .	94
5.4.2	Self Organizing Maps . . . . .	97
5.5	Utilizing Information about Design Concepts . . . . .	102
5.6	Application to Passenger Car Design . . . . .	103
5.6.1	Tree Induction for Car Concept Retrieval . . . . .	103
5.6.2	Car Concept Learning with SOMs . . . . .	107
5.7	Interim Summary . . . . .	112
<b>6</b>	<b>Interaction Analysis</b>	<b>113</b>
6.1	Design Interaction . . . . .	113
6.2	Fundamental Information Concepts . . . . .	115
6.3	Interaction Information . . . . .	118
6.4	Three-way Interaction . . . . .	119
6.4.1	Synergy . . . . .	120
6.4.2	Redundancy . . . . .	122
6.4.3	Non-Interaction . . . . .	124
6.4.4	Compensation . . . . .	125
6.5	Interaction Graphs . . . . .	126
6.6	Utilizing Interaction Information . . . . .	128
6.7	Examples . . . . .	129
6.7.1	Information Normalization . . . . .	129



6.7.2	Compensation of Redundancy and Synergy . . . . .	132
6.8	Application to Passenger Car Design . . . . .	134
6.9	Interim Summary . . . . .	137
<b>7</b>	<b>Summary and Outlook</b>	<b>139</b>
7.1	Summary . . . . .	139
7.2	Outlook . . . . .	141
<b>A</b>	<b>Design Sensitivity Analysis</b>	<b>145</b>
A.1	Direction of performance improvement . . . . .	145
<b>B</b>	<b>Related Publications</b>	<b>147</b>





# 1

## Introduction

### 1.1 Motivation

---

Computer Aided Engineering (CAE) technologies have become powerful tools especially but not limited to the design, evaluation and verification of technical systems. The modeling of physical processes and configurations on computer systems constitutes a major basis for an efficient product development process and for the solution of a huge variety of engineering problems. Technologies for Computer Aided Design (CAD), Computational Fluid Dynamics (CFD) simulation, Finite Element Analysis (FEA), computational optimization and response surface methodologies are well established in the domain of engineering design, and are gradually replacing expensive physical modeling processes. An increase of data resulting from this boost in the usage of computational engineering tools together with high pressure toward faster and efficient product development cycles makes a consistent and thorough information and knowledge handling imperative.

More recently, technologies from computational intelligence and data mining have been adopted to exploit experimental design data and computational resources for the support of engineers in the decision making process. However, the multidisciplinary characteristics of complex design processes and the huge variability in computational design representations hinders the analysis of design data beyond individual design configurations and processes. Especially the variation in the computational representations being used, makes an efficient knowledge exchange between various design processes difficult.

The fusion of information from different computational models becomes indispensable for implementing a holistic data mining process and goes beyond the storage and administration of raw design data. It requires the definition of a unified and typically high dimensional system design representation. Such a representation makes designs exchangeable between design processes and engineers and builds the basis for a holistic analysis of the design data.



## CHAPTER 1. INTRODUCTION

---

Deploying sophisticated technologies from statistics, information theory, data mining, and computational intelligence upon a unified object representation is assumed to be the key paradigm for improving the efficiency of the design process, and for reducing the time to market for products in the future. With the integration of data mining technologies on a unified basis, decisions and knowledge can be made indelible throughout the existence of the developed design and beyond potential reorganizations of design processes, engineering teams and competences.

However, the application of data mining or related technologies to a high dimensional unified representation goes beyond the usage of modern data modeling techniques, it requires the consideration of all aspects of the data mining process including pre-processing, feature extraction, data modeling, visualization and utilization of the considered design data.

The concepts for *shape mining*, introduced in this thesis, instantiate unstructured surface meshes as a unified representation of the shape of three dimensional objects related to the domain of aerodynamic and structural design. The described concepts focus on the investigation of technical systems designed for optimal fluid dynamics performance, motivated from the request of engineers in the domain of car, airplane and boat design. Techniques and prerequisites for integrating technologies regarding the analysis of shape data and the support for decision making within the engineering design process are studied. The suggested methodologies consider the representation of designs, the extraction of valuable information from existing design and performance data and the utilization of the acquired information.

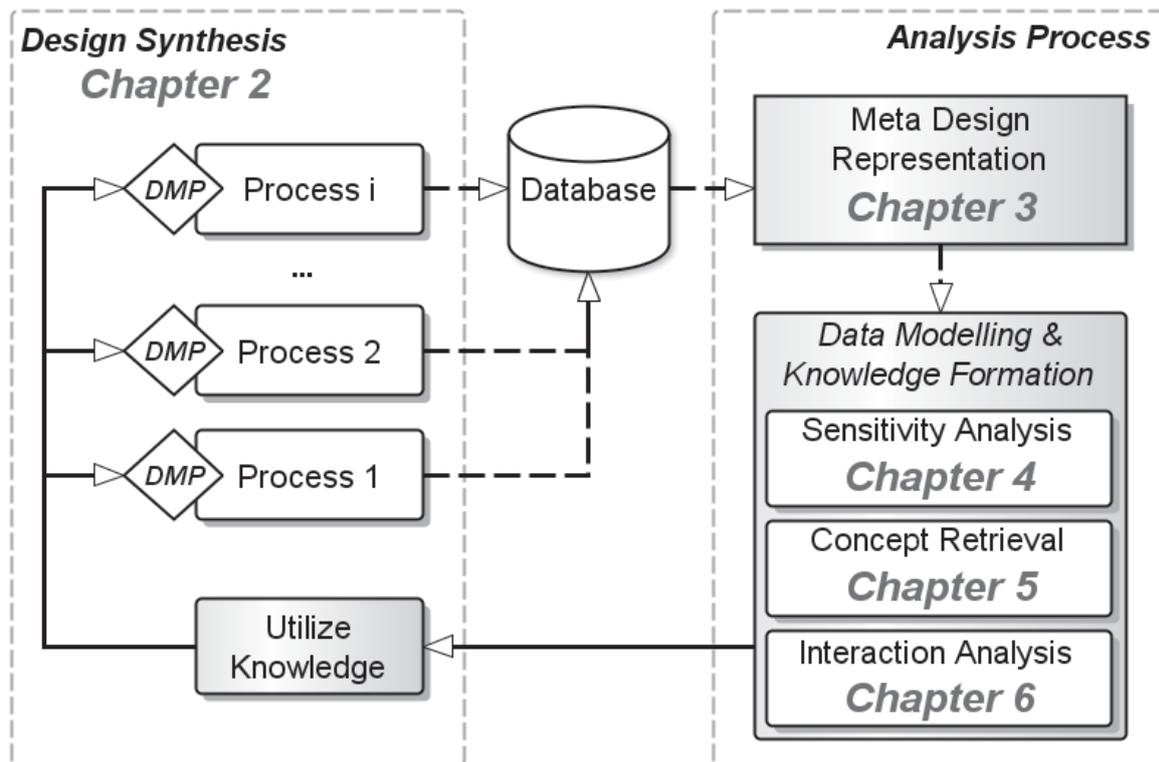
### 1.2 Overview

---

An overview of the *shape mining* concept is depicted in Fig. 1.1. The diagram summarizes the studied topics of this thesis, which is organized as follows.

**Chapter 2** provides an introduction to the engineering design synthesis process. The interplay between engineers, design synthesis and analysis are discussed on a conceptual basis. Thereafter, a compact review of the state of the art in computer aided engineering and design analysis is given. As an example, aspects of the synthesis process are discussed based on the generation of passenger car design data. The resulting designs and performance numbers are the basis for experimental studies throughout the thesis.

**Chapter 3** introduces unstructured surface meshes as a unified object representation in the domain of aerodynamic and structural design. The definition of a unified object representation is a pre-requisite for the implementation of a holistic data modeling process. Methods for evaluating



**Figure 1.1:** *Low level view on the shape mining process. Summary of the main components of the thesis.*

the differences between local surface characteristics are described. Based on the surface mesh representations of the passenger car designs, a statistical analysis of the surface feature variations provides means, e.g., to compare individual designs or to evaluate the course of design processes.

**Chapter 4** studies methods for extracting knowledge about the interrelation between design feature variations and changes in the design performance based on the unified shape representation. Standard methods for sensitivity analysis are reviewed and a robust variant of the mutual information is compared to its information theoretic definition. Applied to the passenger car design data, the sensitivity analysis provides tools to filter potentially irrelevant design features from the design data. K-nearest neighbor algorithms are studied, allowing to control the locality of the sensitivity estimates. The application of more sophisticated methods of knowledge formation necessitates to resolve the typical drawback of the universal design representation, i.e., its high dimensionality. Exploiting the geodesic distance together with the sensitivity information, a new algorithm is proposed to derive larger sensitive design areas, and with that derive a reduced object shape representation.



## CHAPTER 1. INTRODUCTION

---

**Chapter 5** investigates the retrieval of abstract design concepts. The reduced feature sets from the sensitivity analysis facilitate the application of enhanced modeling techniques from data mining and computational intelligence to the design data. In this chapter, the procedure for the retrieval, representation and evaluation of abstract design concepts is generalized and can be carried out independently of the used modeling technique. A new measure of concept relevance is defined, which evaluates extracted design concepts based on the estimation of their utility. The new measure allows the ranking of concepts according to the formulation of the engineers' objectives. Self-organizing maps and decision tree models are studied and applied to the passenger car design data with the purpose of retrieving relevant design concepts described by human-readable design rules.

In **Chapter 6** an information theoretic approach for the identification of design interactions is investigated. The measure of interaction information, an extension of the mutual information to multiple variables, provides a computational means to quantify the joint influence of distant design areas on the design performance. The resulting interaction graph, a graphical representation of the information measures, provides a fast and easy way to visualize rather complex statistical dependencies. Results from the interaction analysis provide valuable information, which can be utilized to decompose complex design tasks into simpler, functional independent subcomponents.

The major results and contributions of this thesis are summarized in **Chapter 7**. Ideas and possible directions for future research are proposed in the outlook of the thesis.



# 2

## Design Synthesis

The engineering design synthesis is a versatile process with a close cooperation between engineers, experimental facilities and computer systems, which furthermore requires an efficient interplay between design and analysis activities, conceptually discussed in this chapter. In the recent years, many computational tools and methodologies have been developed to support engineers in the design, optimization and analysis of technical systems. A compact overview on the state of the art methods in computer aided engineering design are given in this chapter.

The generation of computer readable design data during the design synthesis, e.g., using computer aided engineering tools, is the groundwork for a holistic data driven analysis and knowledge extraction framework. Given the exterior design of a passenger car as an example, relevant properties of the design process are explained. Throughout the thesis, the generated passenger car design data defines a solid base for studying data mining concepts for knowledge extraction within the shape mining framework.

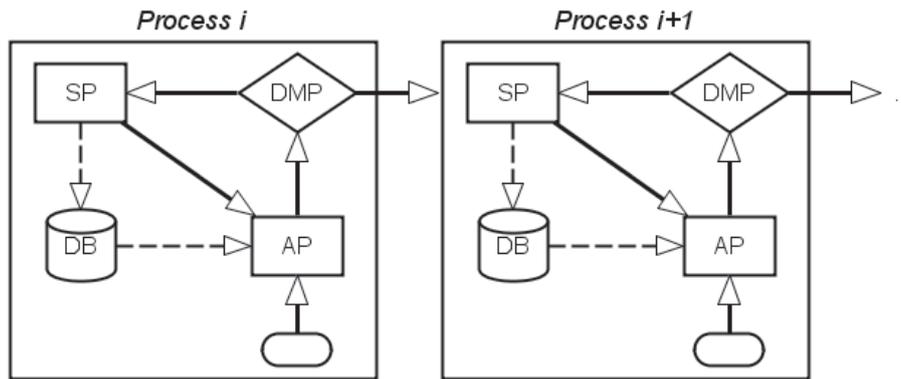
### 2.1 Engineering Design

---

The engineering design process can best be described as a goal oriented iterative decision making process [Pahl et al., 2007; Evbuomwan et al., 1996]. In each iteration, engineers decide about individual or a sequence of design variations that lead to a final design configuration, fulfilling pre-defined constraints and design goals. This requires that initial decisions about the design representation, the evaluation technique and constraints are made, which are continuously reviewed during the progress of the design process. The efficiency and success of any design process depends on the decisions taken and thus on the experiences and knowledge of the engineers. In order to augment and distribute any supplementary knowledge, a distinct analysis process is indispensable after the actual modeling, production and evalua-

## CHAPTER 2. DESIGN SYNTHESIS

tion of design instances. As depicted in Fig. 2.1, each design process can be partitioned into distinct phases: the actual *synthesis process*, referring to the process of generating new design variations, the *analysis process*, relating to the formation of new domain knowledge from the results, and finally the *decision making process* that derives new formulations about the design strategy, representation, constraints and objectives.

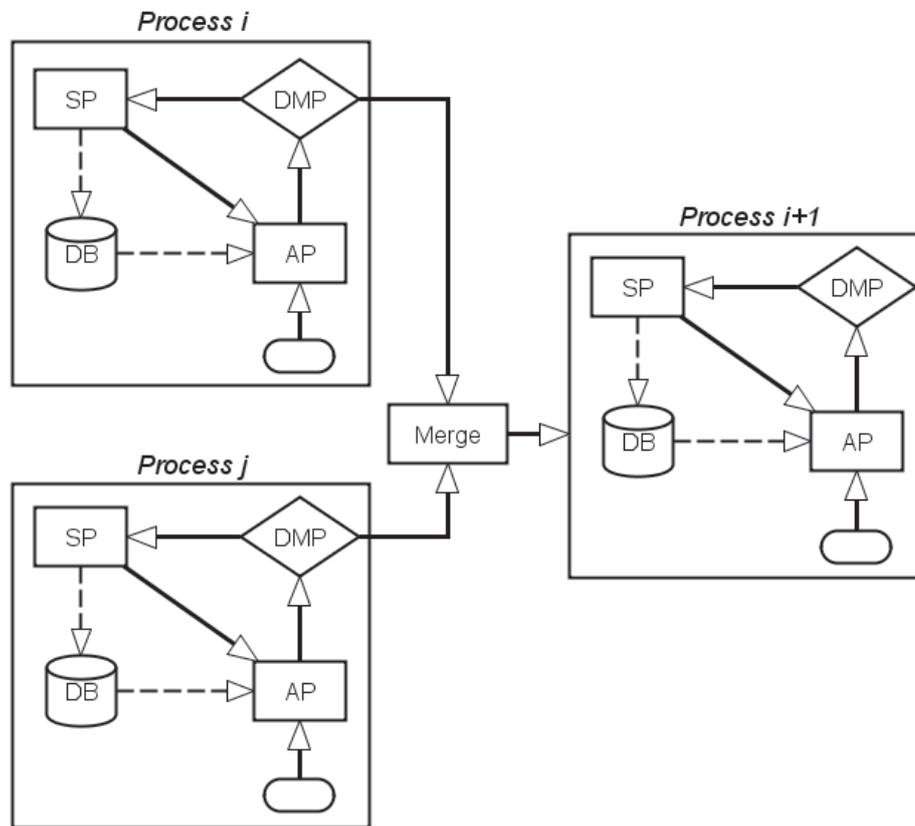


**Figure 2.1:** Flow chart of a serial design process configuration. Process  $i$  defines the baseline of the subsequent process  $i + 1$ , where design information is passed from process  $i$  to its successor. Each process includes the actual synthesis process (SP), the analysis processes (AP), the decision making process (DMP) and a database (DB) for data storage.

Large scale systems often require a subdivision of the entire design configuration and the design process in order to handle their complexity. Thus, the outcome of each design process defines just an intermediate step within the overall synthesis process. Such interrelations between processes can be classified into serial or parallel process configurations, illustrated in Fig. 2.1 and Fig. 2.2, respectively.

In a serial configuration, preceding design processes define the starting configuration or constraints for a subsequent design process. Process information is transferred at the interface between successive processes. Typically, the exchange of information is limited to information related to the final design configuration. The exploitation of intra-process information from process  $i$  by process  $i + 1$  is not modeled in such a pattern and completely relies on the engineers involved. Thus, design data and the results from the analysis and decision making are properties of each individual process, where in each process often distinct representations of the design or parts of it are used. Hence, variations are applied and decisions are made based on individual representations, ignoring the holistic view on the design. The analysis

process, where decisions are made, refers to the representations of the design in the process and does not integrate information, e.g., from preceding processes.

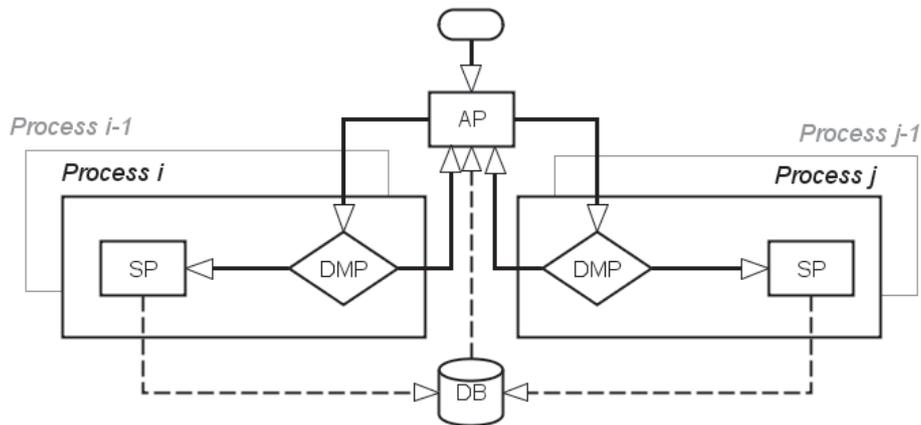


**Figure 2.2:** Flow chart of a parallel process configuration. Results of parallel processes  $i$  and  $j$  need to be merged before starting a subsequent process  $i+1$ . Each process includes the actual synthesis process (SP), the analysis processes (AP), the decision making process (DMP) and a database (DB) for data storage.

In a parallel design configuration a design process  $i$  is split into parallel sub-processes  $i$  and  $j$ , as depicted in Fig. 2.2. The parallelization of individual design processes can increase the efficiency of the overall process. However, results of parallel processes need to be merged before starting any subsequent process. Thereafter, design information is exchanged after finalizing all parallel processes. A design configuration which is superior with respect to the constraints and targets of one design process  $i$  might be inferior in the light of a second concurrent process  $j$ . In an extreme scenario, different parallel design processes might even act against each other, especially when each

## CHAPTER 2. DESIGN SYNTHESIS

process targets the design of a distinct part of the holistic design. Merging the results of parallel processes can sometimes lead to surprising observations with respect to overall design properties. Additional expensive subsequent design steps have to be carried out to balance possible debasements.



**Figure 2.3:** *Flow chart of the integrative design processes configuration. The synthesis process (SP) and the decision making process (DMP) are properties of individual processes, whereas the analysis process (AP) integrates information from all concurrent and serial processes, requiring the instantiation of a shared database (DB).*

Both, serial and parallel configurations are found in nearly any real world design process, even so they are not implemented in its pure form. Processes might partially overlap. Furthermore, serial and parallel process patterns are used in combination. However, the problems described above still hold. Analysis and decision making processes are mostly a property of individual design processes.

For an efficient co-evolution of design processes, design data and related information needs to be accumulated beyond individual processes throughout the course of the entire design synthesis, by means of implementing an integrative process configuration. The concept of the integrative process configuration is depicted in Fig. 2.3. The implementation of a shared database together with a unification of the design data, i.e., the definition of a unified design representation and the unified evaluation of the design performance, facilitates the implementation of a holistic analysis process. In such a configuration, individual design decisions can be made in consideration of the holistic design. Such global and holistic view on the design data allows an integrated exchange of domain knowledge at any stage of the design pro-



cess. Thus, a late merging of process results could be omitted, increasing the efficiency of the overall process.

To exploit the potential of the integrative process configuration, methods from the Computer Aided Engineering (CAE) and Data Mining domain need to be utilized to aid the analysis and design synthesis process. Furthermore, new attempts for the holistic analysis of the design data need to be established.

## 2.2 Computer Aided Design Synthesis

---

Engineers involved in the synthesis of new designs are supported by a huge assortment of Computer Aided Engineering (CAE) tools, which aid, e.g., the modeling, creation, evaluation and optimization of technical systems. With the introduction of Computer Aided Design (CAD) systems, engineers moved from their writing desk to computerized workstations [Keane and Nair, 2005]. Initially designed for drafting, nowadays, modern CAD systems capture information about geometric and material properties, as well as process and manufacturing information for individual designs. From the specification of the design shapes, high fidelity simulation tools for Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEA) facilitate the evaluation of the functional characteristics of each design on the computer, step by step replacing expensive physical experiments. With the support of advanced visualization technologies, domain experts can profit from the analysis of individual simulation results by accumulating information for consolidated decision making. Based on the design constraints, the objectives of the design process and the result of individual simulations, engineers decide about successive design configurations.

While the decision making based on individual design and evaluation results is a demanding process, reflecting a trial and success attempt, methods for the Design of Experiments (DOE) [Forrester et al., 2008; Morris and Mitchell, 1995] allow a more systematic approach to explore feasible design configurations. On the basis of the DOE results, response surface methodologies [Myers et al., 2009] are applicable to speed up the experimental design process by identifying potentially outperforming design configurations. The combination of computational optimization technologies with high fidelity simulations for CFD simulations or FEA can be seen as an additional big step towards the automation of design processes. Given a set of pre-defined constraints and objective settings, and typically given a fixed object representation, computational optimization technologies implement heuristics to imitate the iterative decision making process on computers, see [Keane and



## CHAPTER 2. DESIGN SYNTHESIS

---

Nair, 2005] for an introduction. Nature inspired, derivative free search strategies like genetic algorithms [Holland, 1992], evolution strategies [Rechenberg, 1971; Beyer and Schwefel, 2002], simulated annealing [Kirkpatrick et al., 1983] or particle swarm optimization [Kennedy and Eberhart, 1995] became prominent in the domain of engineering design in the recent years. An extensive overview concerning such optimization technologies is provided by [Montaño et al., 2012] or [Saridakis and Dentsoras, 2008].

With the selection of the computational optimization algorithm, and with the specification of the representation, optimization objectives and constraints, the engineers define the strategy for each individual design process. Computer aided optimization methods, e.g., the CMAES (Evolution Strategy with Co-variance Matrix Adaptation) [Hansen and Ostermeier, 2001] or model guided evolution strategies [Graening et al., 2010; Reehuis et al., 2011] even adapt their search strategy online depending on the current state of the search process. As shown in [Olhofer et al., 2001], an online adaptation of the design representation during the search does relax the initial search constraints and can lead to new outperforming design solutions.

In engineering design optimization the evaluation of the objective functions often requires to conduct computational expensive high fidelity simulations, limiting optimization and search strategies to unfold their theoretic potential. Surrogate assisted search strategies, as the result of the progress made in the domain of artificial intelligence, allow a partial replacement of the expensive fitness evaluations by fast approximation models during the progress of the optimization, e.g., see [Jin, 2005; Forrester et al., 2008; Jin, 2011] for an overview.

Each search and optimization process aided by computer systems produces a multitude of computer readable design data. Beyond the optimized design configuration, each design and its related performance carries important information about the design domain. The extraction of valuable information from all the generated design data requires a systematic attempt for data analysis.

### 2.3 Data Analysis

---

The analysis of design data generated is an indispensable step in any design process. A precise data analysis may produce new domain knowledge, which can support engineers in decision making. Methods from data mining provide a systematic formalism and the necessary means for the analysis of stocks of data.

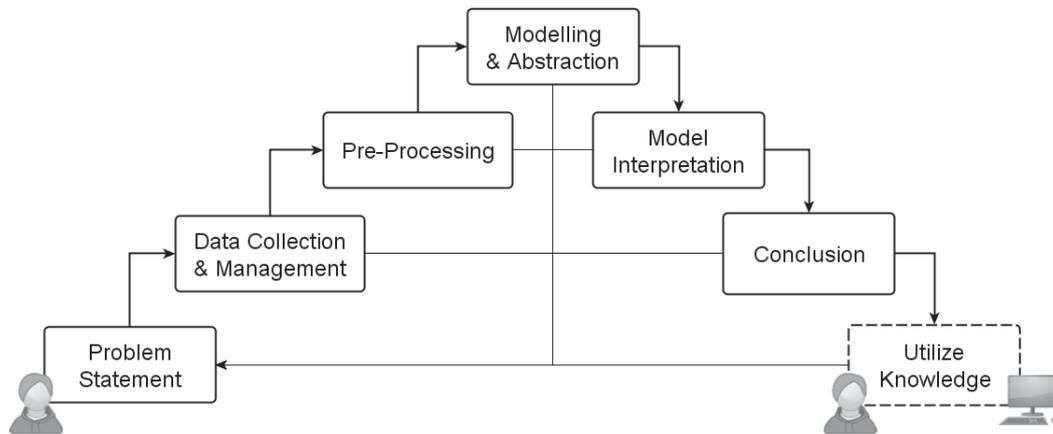
### 2.3.1 Data Mining

Data mining is a relatively young and interdisciplinary research field in computer science, which constantly rises in importance, see [Kantardzic, 2001; Rokach and Oded, 2005]. Nowadays, engineers, scientists, biologists, economists and others profit in their daily business from advances made in this field. Data mining originates from classical data analysis by means of statistics and machine learning and integrates techniques from various disciplines like evolutionary computation, artificial intelligence, fuzzy logic, rough set analysis, information theory, and modern data management, to name just a few. Without attempting to capture all aspects, Kantardzic in [Kantardzic, 2001] provides a short definition of the term data mining: "Data Mining is a process of discovering various models, summaries, and derived values from a given collection of data".

The interdisciplinarity of data mining is based on the fact that data mining is strongly driven by the posed question to be answered by the given data, rather than based on certain universal principals observed in nature. Roughly speaking, data mining provides a standardized procedure together with a huge collection of methods and concepts for efficiently approaching the investigation of large amounts of data. Practical experience reveals an important point, which was explicitly raised by [Kantardzic, 2001], data mining is not solely the selection of an appropriate computer-based tool for a given problem. Rather, data mining needs to be understood as an iterating process, which repeatedly revisits the problem stated, the methods chosen, the results obtained and the conclusions drawn.

According to [Kantardzic, 2001], a typical data mining application needs to consider the implementation of distinct data mining steps, namely: the problem statement, the data collection, the data pre-processing, the modeling of the data, the interpretation of the models, and finally a step for drawing conclusions from the interpretations made. Different data mining processes with a varying number of individual data mining steps have been introduced, see e.g. [Han and Kamber, 2006] or [Rokach and Oded, 2005]. However, conceptually they all suggest the implementation of similar individual steps. The utilization of the conclusions drawn or the knowledge retrieved is seldom discussed explicitly in the data mining literature but is considered an important issue in any data mining process.

Fig. 2.4 summarizes all steps of the data mining process that are considered to be relevant, mainly following Kantardzic's definition of the data mining process [Kantardzic, 2001], enhanced with a step for utilizing the acquired knowledge. Each of the considered steps will be shortly described.



**Figure 2.4:** *Data mining process according to Kantardzic, see [Kantardzic, 2001].*

The *problem statement* is one of the most important steps in the entire data mining process that strongly depends on the application domain. It requires to get a thorough understanding of the application and the objectives of the overall data mining process, so that the individual data mining steps can be clearly defined. Very often, the problems are related to the identification of the intrinsic data structure or the discovery of novel patterns in the data. In the process of formulating the data mining problem, it needs to be considered, that the information that can be extracted can only be as rich as the underlying data and data representation allows.

For an efficient *data collection and management*, sophisticated *database management systems* (DBMS) are available. Nowadays, even open source database systems provide professional tools to organize, store and retrieve domain specific data. Nevertheless, it needs to be guaranteed that the stored data is consistent, where the definition of standards and unified data formats can help to keep this consistency.

In the *pre-processing step* meta data required for the actual data mining process needs to be selected from the entire dataset. It can further be of avail to transform the data into a different representation, or to handle missing values and remove outliers. Already the implementation of an appropriate pre-processing step can be the major challenge for the realization of the entire data mining process, see [Cios and Kurgan, 2005]. Spending more effort on an appropriate pre-processing can ease the subsequent data modeling significantly.

The *modeling step* targets to derive a computational model from a given set of data. In data mining one distinguishes between descriptive and predictive models. While descriptive models build up human readable representations of the data that target to identify new patterns and explain underlying

phenomena, predictive models aim to approximate the data to predict outcomes of previously unseen data samples. A rich set of modeling techniques is available which all have their advantages and drawbacks. A comprehensive summary of modeling techniques is given by [Rokach and Oded, 2005]. Aiming to select the right model, data mining experts face the trade-off between interpretability and accuracy. Data mining techniques are applied to support decision making, which requires that conclusions can be drawn from the generated model. Therefore, the most accurate model might not always be the best choice to fulfill the goal of the data mining process.

The step of *interpreting the model* is aiming to transfer the model into a human readable knowledge representation. The abstract knowledge represented by the data models needs to be prepared and presented to the application experts, from which *conclusions* can be drawn. Meaningful conclusions from the data mining process can often be made only in a close cooperation between data mining and application experts.

The *utilization* of the acquired knowledge is often not considered explicitly in the data mining process, but is an important step, especially in the engineering design process. In the utilization step, based on the conclusions drawn from the model, a certain strategy needs to be defined to use the resulting knowledge for the improvement of subsequent design processes.

### 2.3.2 Data Mining for Aerodynamic and Structural Design

Since the amount of computer readable data generated during design processes increases, data mining can be an important technology to exploit the resulting engineering design data. However, literature related to the extraction of human readable knowledge in the field of aerodynamic and structural design is rare. In the following, the most relevant contributions are reviewed.

In [Obayashi and Sasaki, 2003; Jeong et al., 2005; Chiba et al., 2005; Obayashi et al., 2005, 2010], the authors have addressed the need for the extraction of knowledge from a given set of design data to gain insights into the relationship between geometry and multi-criteria performance measurements. The authors applied self-organizing maps (SOM) in order to find groups of similar designs for multi-criteria performance improvements and tradeoffs, and used the analysis of variance (ANOVA) technique to identify the most important design parameters. For example, their methods have been applied to supersonic wing design.

In [Graening et al., 2009; Rath and Graening, 2011], methods from information theory for the purpose of interaction analysis have been investigated.



## CHAPTER 2. DESIGN SYNTHESIS

---

With that, the authors targeted to reveal higher order interrelations between design and flow field properties, tested in the domain of turbine blade and passenger car design.

In the common approaches, the extracted information is always linked to a specific and well-defined design representation. Thus, the usability of the extracted information beyond a particular design and optimization process is limited. To overcome this drawback, in [Graening et al., 2008] the use of data mining techniques on the basis of a unified design representation has been studied. It has been recognized that the extraction of knowledge based on a uniform design representation goes beyond the application of individual modeling technologies. It requires the consideration of many aspects of the complete data mining process.

### 2.4 Passenger Car Design Synthesis

---

During the design synthesis of complex systems, typically, design data evolves from various diverse design processes, where each design process follows a pre-defined strategy to reach a specific design goal. In this section two design strategies, as they are often used in CAE, are carried out to create optimal shapes of a passenger car. The first one implements a global search strategy by means of uniformly sampling a constrained design space. The second strategy follows a direct local search attempt exploiting the characteristics of the designs during the progress of the design process. Both strategies result in design datasets which are characteristic for an explorative and an exploitative design and optimization process. Typically, a sensible combination of the two strategies is used, both for computational as well as human driven engineering design. The design space, which is representing all potential passenger car shapes is restricted to their geometrical representation. In this study variations of the passenger car shape are modeled using Free Form Deformation (FFD). The improvement of the aerodynamic performance of each passenger car is pursued, defining the overall design goal based on the results from computational fluid dynamics simulations (CFD).

#### 2.4.1 Design Representation

To model variations of a passenger car, Free Form Deformation (FFD) [Sederberg and Parry, 1986] has been applied to the initial car shape. Originating from the field of computer graphics, FFD has become a valuable technology for shaping designs in the recent years. In contrast to most parametric design representations, FFD represents variations of a chosen baseline design.



## 2.4. PASSENGER CAR DESIGN SYNTHESIS

---

Depending on the setup of a control point grid relative to the embedded objects, it allows to apply global and local deformations similarly. Applying the FFD representation to the car shape requires to setup a three dimensional control point grid, which embeds the point based representation of a baseline car. The control points of the grid serve as handles for the deformation of the embedded objects. The parameterized control grid defines the degrees of freedom and constraints for the respective design process.

The choice of the representation strongly depends on the target setting of each individual design process. On one hand, if the representation severely limits the degrees of freedom one might fail in reaching the desired design goal. On the other hand, if the flexibility of the representation is too high the search process can fail to discover an optimal design due to the high dimensionality of the search space. As a consequence, the representation seldom remains unchanged during the entire synthesis of a new design. Typically, different variations or conceptually different representations are applied until all design goals are met.

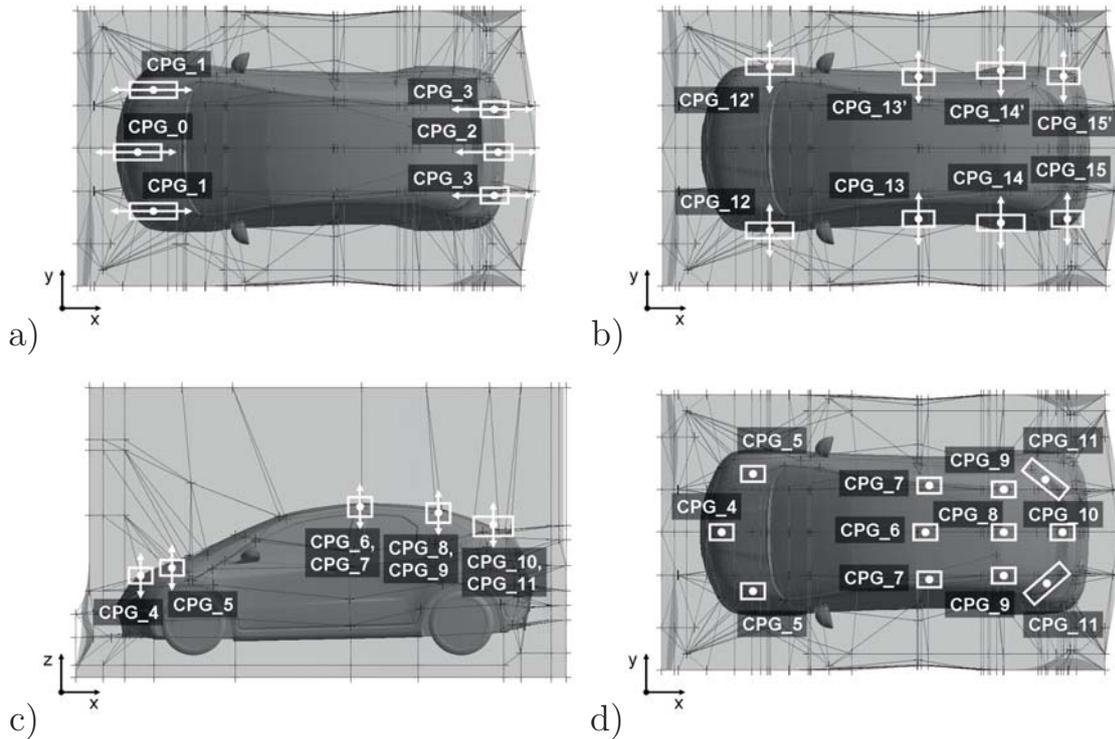
As an example, two different control point grids have been constructed. The first one, representation A, incorporates expert knowledge by means of introducing design constraints, which ensure a certain practicability of the car. The second control point grid, representation B, implements a standard setup without the consideration of any a priori domain information. Both grids embed the surface of the baseline car shape that is represented by a triangular surface mesh. All vertices of the surface mesh are "frozen" to the respective control points. The freezing step transfers all object points from the Cartesian into a trivariate spline coordinate system. For the setup of the control point grid, the freezing process and the deformation of the baseline shape *VisControl*<sup>1</sup>, a comprehensive in-house software has been used. After the mesh has been frozen, a displacement of the control points results in a deformation of the embedded shape according to the spline definition. Further details related to the theory of FFD are provided by [Sederberg and Parry, 1986].

### Representation A

The first FFD representation is represented by a control grid consisting of  $m_A = 567$  control points,  $\mathcal{P}_A$ . Wherein, splines of degree 3 (order 4) are utilized. A significant portion of the control points has been introduced to constrain the deformations on the baseline mesh, further denoted as  $\mathcal{M}_I$ . For example, fixed control points have been introduced along the wheel

---

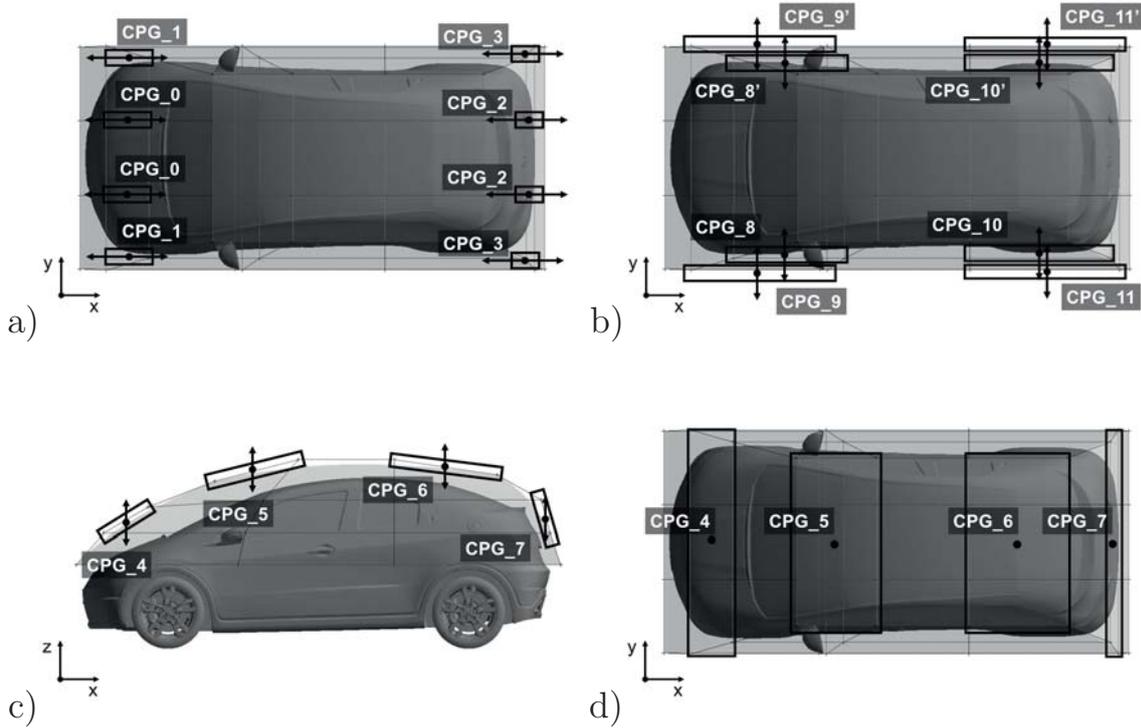
<sup>1</sup>Developed at the Honda Research Institute Europe GmbH (HRI-EU)



**Figure 2.5: Representation A:** Setup of the control point groups for inducing modifications on the passenger car surface. 16 control point groups are built to deform the surface into the  $x$  (a),  $y$  (b) and  $z$  (c,d) direction. This representation has in particular been tuned for an realistic optimization of the car shape using expert knowledge.

house and the front window to restrict modifications at respective surface regions. Such limitations ensure, e.g., the manufacturability of the resulting car designs. Based on the control volume,  $k_A = 16$  control point groups,  $\mathcal{G}_A = (CPG_0, \dots, CPG_{k_A-1})^T$ , have been defined. The individual control point groups are marked and labeled in Fig. 2.5 a) to d). The different figures represent different axial-parallel projections on the same 3D control point grid. The displacement of the control points within each group is restricted to displacements along individual axes. The control points of groups  $CPG_0$  to  $CPG_3$  are restricted to modifications in  $x$  direction. The groups  $CPG_4$  to  $CPG_{11}$  and  $CPG_{12}$  to  $CPG_{15}$  comprise control points limited to the modification in  $z$  and  $y$  direction, respectively. Furthermore, only modifications are allowed that keep the car design symmetric along the  $y$  direction. All modifications are applied to the chassis of the car only. Wheels are not affected by any modification. All in all, the introduced control point groups define  $k_A = 16$  tunable parameters for defining new car shapes.

## 2.4. PASSENGER CAR DESIGN SYNTHESIS



**Figure 2.6: Representation B:** *Alternative representation of the passenger car shape without using explicit expert knowledge. Illustrations a) to d) visualize the configuration of the control point grid and the individual control point groups for inducing modifications on the passenger car surface. 12 control point groups are build to deform the surface of the upper car chassis into the  $x$  (a),  $y$  (b) and  $z$  (c,d) direction.*

Formally, given variations of  $\mathcal{G}_A$ , representation A defines the mapping from the initial shape  $\mathcal{M}_I$  to a modified shape  $\mathcal{M}'$ :

$$\mathcal{R}_A(\mathcal{M}_I, \mathcal{P}_A, \mathcal{G}_A) : (\mathcal{M}_I, \mathcal{P}_A) \rightarrow (\mathcal{M}', \mathcal{P}'_A), \quad (2.1)$$

with  $\mathcal{G}_A \in \mathbb{R}^{k_A}$  and  $\mathcal{P}_A, \mathcal{P}'_A \in \mathbb{R}^{m_A \times 3}$ .

### Representation B

The second control point grid is a standard representation introduced as an alternative to representation  $\mathcal{R}_A$ . It refrains from a detailed tuning of the control grid. Representation  $\mathcal{R}_B$  results in a control point grid with  $m_B = 64$  control points. In contrast to  $\mathcal{R}_A$ , this representation is restricted to deformations of the upper chassis part. Further constraints, which ensure the feasibility of the designs are not included. As for  $\mathcal{R}_A$  splines with degree 3 and order 4 are used. For the parameterization of the control point grid, control points are effectively grouped into  $k_B = 12$  control point groups,

$\mathcal{G}_B = (CPG_0, \dots, CPG_{k_B-1})^T$ , visualized in Fig. 2.6 a) to d). Again, the modifications of the individual groups are restricted to displacements along distinct axis. The control point groups  $CPG_0$  to  $CPG_3$ ,  $CPG_4$  to  $CPG_7$  and  $CPG_8$  to  $CPG_{11}$  are restricted to displacements in  $x$ ,  $z$  and  $y$  direction, respectively. In summary, given the  $k_B = 12$  variable design parameters, the modification of the initial shape utilizing representation B is defined by:

$$\mathcal{R}_B(\mathcal{M}_I, \mathcal{P}_B, \mathcal{G}_B) : (\mathcal{M}_I, \mathcal{P}_B) \rightarrow (\mathcal{M}', \mathcal{P}'_B), \quad (2.2)$$

with  $\mathcal{G}_B \in \mathbb{R}^{k_B}$  and  $\mathcal{P}_B, \mathcal{P}'_B \in \mathbb{R}^{m_B \times 3}$ . Compared to representation A, the reduced mesh density allows larger variations of the car shape.

### 2.4.2 Computational Fluid Dynamics Simulation

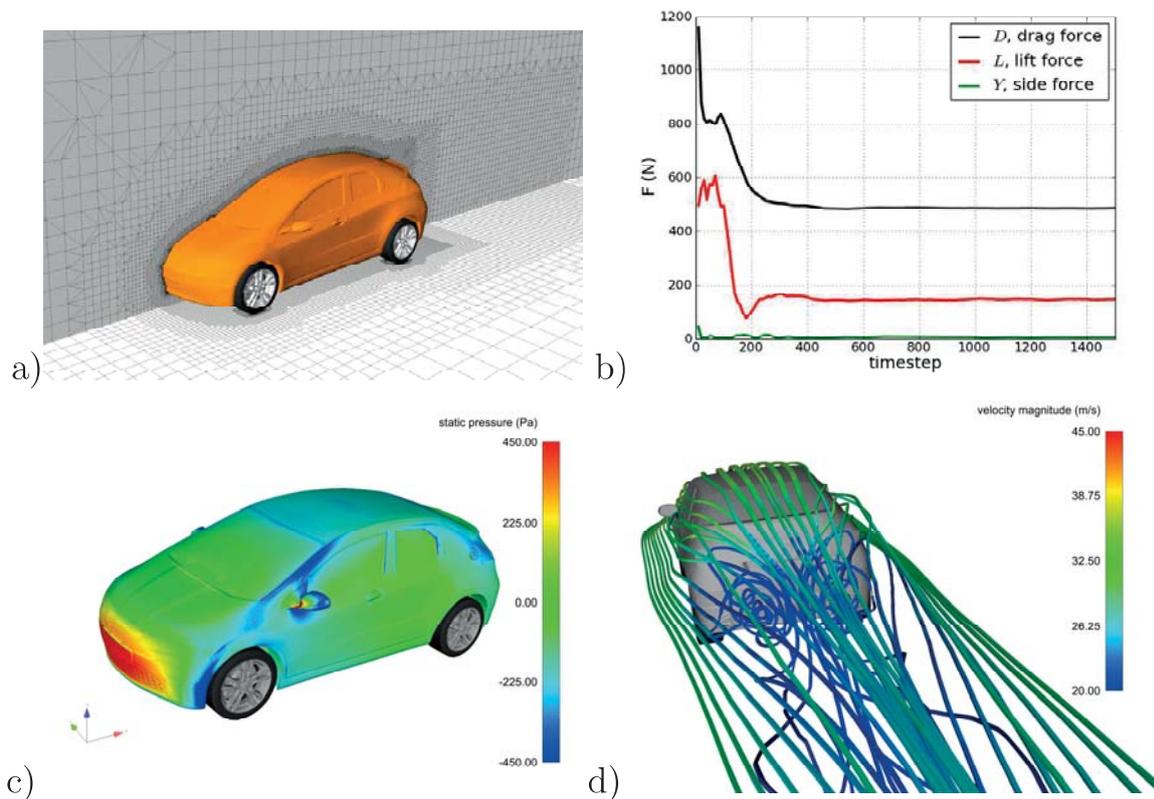
For each design that has been generated using FFD the aerodynamic performance has been evaluated by running a computational fluid dynamics simulation. OpenFOAM<sup>2</sup>, an open source CFD software package is used for simulating the flow around the passenger car surface. The domain occupied by the air flow is subdivided into discrete cells, the CFD mesh. Additionally, boundary conditions are specified, which define the flow behavior at the boundaries of the computational area, e.g., at the inlet or the design surface. Finally, the discrete partial differential equations are solved iteratively until a steady-state is reached.

For the simulation of the passenger car an octree based hexahedral CFD mesh [Schneiders et al., 1996], subdivided into about 3.3 million cells, has been generated from the surface mesh representation using the openFOAM tool *snappyHexMesh*. Fig. 2.7 a) shows the CFD mesh of the baseline design at slices through the three dimensional mesh. The mesh enfolds a volume of  $153.0m \times 51.2m \times 26.0m$ . A uniform flow with a velocity of 110 km/h at the inlet of the flow domain is exposed to the car, simulating the car running at the respective speed. According to the simulated velocity, the wheels of the car model are rotating at a speed of 97.53 rad/s. The Reynolds-averaged Navier-Stokes equations are solved including the SST  $k-\omega$  turbulence model, see [Menter, 1993]. The residual of the solution to the partial differential equations is shown in Fig. 2.7 b). The simulation of the flow is parallelized onto four CPU's to reduce computational time. The overall computational time for one simulation of the flow around the full passenger car model, including the generation of the CFD mesh, is about 3h relating to a four core Xeon E5620, 2.4GHz processor. The resulting static pressure distribution along the surface of the simulated initial car shape is shown in Fig. 2.7 c),

---

<sup>2</sup><http://www.openfoam.com>

## 2.4. PASSENGER CAR DESIGN SYNTHESIS



**Figure 2.7:** Illustration a) shows 2d slices through the hexahedral volume mesh to provide a qualitative impression of the CFD mesh characteristics. The residuals from solving the Reynolds-averaged Navier-Stokes equations are depicted in b). Illustration c) and d) show the typical flow characteristics and pressure distribution around the bodywork of a passenger car.

allowing to identify areas of high and low pressure directly linked to the overall flow resistance. In order to get a visual impression of the actual flow around the car shape, selected stream lines around the rear part of the car are shown in Fig. 2.7 d). The visualization unveils a typical vortex structure resulting from a flow separation at the rear of the car. The elongation and structure of the vortex has a major effect on the overall drag [Hucho, 2003].

Mainly two quantities are derived from the solution of the flow simulation, the overall drag force  $F_D$ , acting on the car in the direction of the freestream flow, and the rear lift force  $F_{LR}$ , which is perpendicular to the fluid flow.  $F_D$  and  $F_{LR}$  are used to quantify the aerodynamic characteristics of each passenger car model. Typically, engineers target the reduction of  $F_D$  and  $F_{LR}$  at the same time, since  $F_D$  is directly linked to the fuel consumption and the reduction in  $F_{LR}$  to an increase in the car stability.

### 2.4.3 Explorative Design with Latin Hyper Cube Sampling

Sampling methods are widely used in engineering design in order to explore a constrained design space. Optimal sampling strategies are highly relevant for applications where full factorial experiments are infeasible due to high experimental costs and linked with constraints on the available resources. In engineering design, sampling plan methods are typically applied in order to produce properly distributed data for constructing an approximation of the quality function, see [Forrester et al., 2008]. Subsequent optimization and design processes can utilize these approximation models as surrogate for expensive quality function evaluations. Without the demand on naming all existing sampling techniques, the most prominent and most frequently used sampling techniques are *random Latin hypercube sampling (LHS)*, *space-filling LHS*, *Sobol sequences* or *orthogonal arrays*.

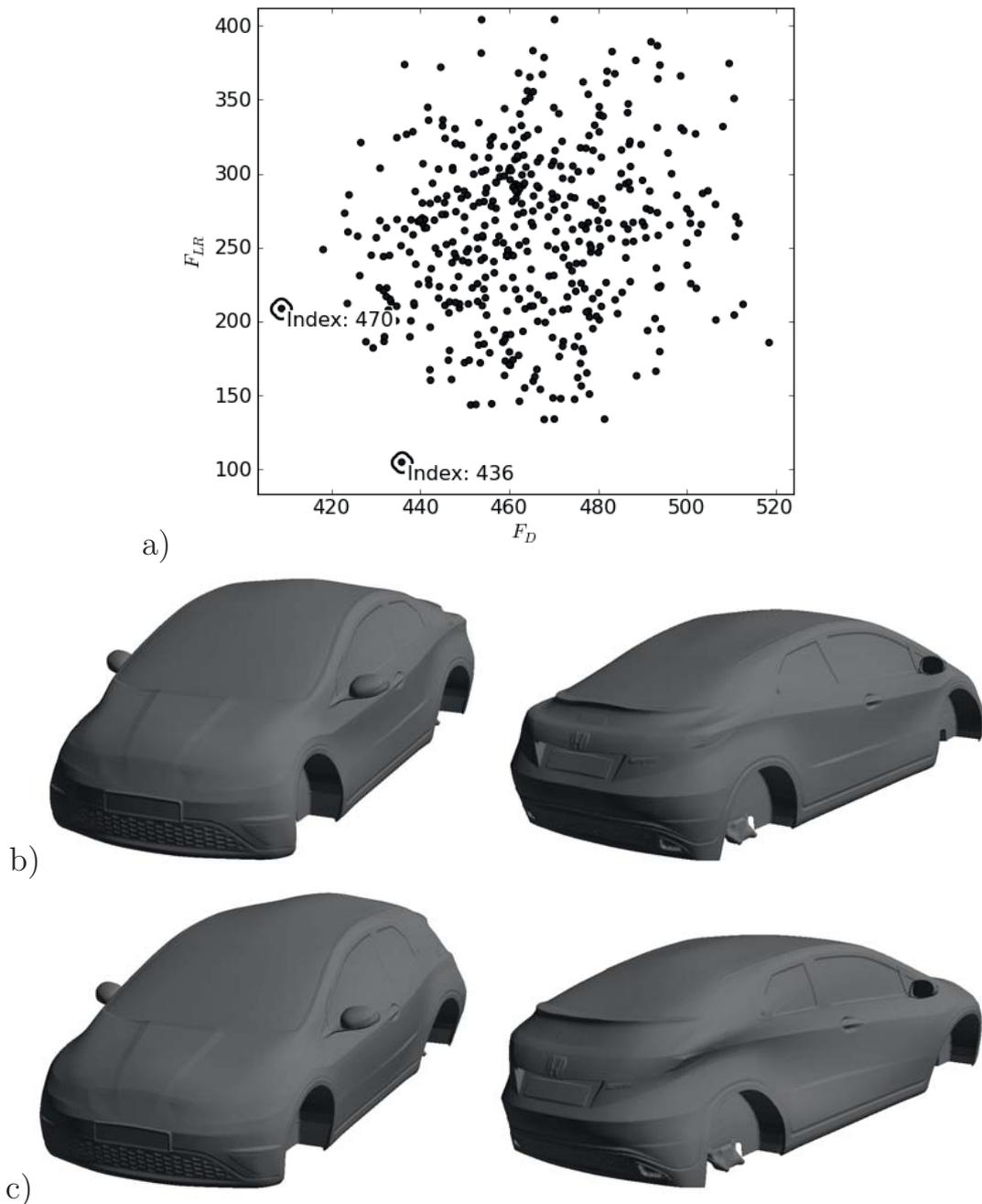
In this study, an optimized LHS method, as described in [Forrester et al., 2008], is used. The optimized LHS method applies the optimality criteria of Morris and Mitchell, defined in [Morris and Mitchell, 1995], to achieve a space-filling sampling. The generated design dataset is seen to be representative for all explorative space-filling sampling methods. The most relevant properties of the optimized LHS technique are shortly reviewed. Given the constrained design domain, further referred to as  $\mathcal{D}$ , normalized onto the  $k$ -dimensional unit cube  $\mathcal{D} = [0, 1]^k$ , the target is to discover an optimal sampling plan  $\mathcal{X}$  that contains  $n$  sample points uniformly distributed in  $\mathcal{D}$ . To achieve a uniform sampling, the design space  $\mathcal{D}$  is partitioned into a rectangular grid of  $n^k$  elements. Latin hypercube sampling (LHS) does aim at a stratified sampling, meaning that sampling points are generated whose projection onto the individual dimensions are uniformly distributed. However, fulfilling the stratification criteria solely does not guarantee a uniform distribution of sample points within the entire design space. Therefore, an additional optimality criterion is required to ensure a uniform space-filling. In [Morris and Mitchell, 1995], Morris and Mitchell provide a max-min metric which allows to rank sampling plans according to their uniformity,  $\Phi_q(\mathcal{X}) = \left( \sum_{j=1}^m J_j d_j^{-q} \right)^{1/q}$ , where  $d_1, d_2, \dots, d_m$  defines the distances between all possible pairs of points in  $\mathcal{X}$ , sorted in ascending order, and  $J_1, J_2, \dots, J_m$  being the number of pairs of points in  $\mathcal{X}$  separated by the distance  $d_j$ . As distance measure  $d_j$  the Manhattan distance has been used. The scalar  $q$  is a free parameter of the metric. Given the uniformity metric  $\Phi_q$  facilitates the use of computational optimization algorithms to search for an optimal sampling plan. In the present implementation, the search is carried out using an  $(1 + \lambda)$  evolution strategy according to [Forrester et al., 2008]. The authors

utilize swapping of two elements within any columns of  $\mathcal{X}$  as evolutionary operator and perform an automatic adaptation of the number of swaps over each generation.  $\Phi_q$  defines the objective function that drives the search towards the optimal sampling plan. The authors include in their algorithm a heuristic to automatically select a proper value for  $q$ . For more details on the used algorithm the reader is referred to the original paper.

Given the parameterization of representation A, as described in Section 2.4.1, an optimal sampling of the  $k = 16$  dimensional design space is targeted. In order to achieve a sampling density proportional to a density of 5 values in one dimension,  $5^{16}$  experiments would be required within a full factorial setting. Taking into account that one CFD simulation required by each experiment takes about  $3h$  makes the full factorial experiment completely untrackable. Hence, the optimized LHS is applied to generate a constrained number of 500 designs within a constrained designs space. Each dimension is bounded between  $-0.3 \leq \Delta CPG_i \leq 0.3$ , which corresponds to a maximum displacement of each control point group by 0.3 meters. Given a population size of 20 the optimization of the sampling plan is carried out for 100 iterations. The samples are positioned in the center of each hyper cubic element within the discretized design space. Each of the 500 resulting data vectors is transcribed into variations of the control volume grid. Utilizing FFD, 500 modified instances of the baseline design are generated, wherein the variations of the control point grid result in deformations of the baseline surface. For all modified design instances the airflow around the bodywork of the passenger car is simulated and its aerodynamic characteristics are calculated according to Section 2.4.2. The relation between the drag force  $F_D$  and the rear lift  $F_{LR}$  is depicted in the scatter plot of Fig. 2.8 a). A view on the shapes of two different non-dominated solutions are given in Figs. 2.8 b) and c). The generated shape data and the parameter values are stored in a design database and linked to the resulting characteristic values.

### 2.4.4 Exploitive Design with an Evolution Strategy

While sampling techniques target a uniform sampling of the entire design space, optimization algorithms like evolution strategies perform a local sampling along certain paths towards optimal solutions. Optimization algorithms often adapt their strategy parameters by exploiting information about the previously generated solutions. In the following experiments, two optimization runs are carried out targeting the minimization of a pre-defined fitness function. In order to realize the exploitive search process the  $(\mu_w, \lambda)$  evolution strategy with covariance matrix adaptation (CMA-ES) has been used. The CMA-ES optimization strategy has been introduced by Hansen and



**Figure 2.8:** Plot a) depicts a summary of the designs from the explorative search in the objective space. The designs illustrated in b) and c) show the shape of two deformed surfaces (a) index 470, b) index 436) selected from the set of non-dominated solutions.

Ostermeier, see [Hansen and Ostermeier, 2001]. Following the underlying biological concept, design parameters  $\vec{x}$  are coded into the chromosome of the individuals in a population. The so called parent population is initialized

## 2.4. PASSENGER CAR DESIGN SYNTHESIS

with  $\mu$  individuals, which are randomly chosen from the design domain. In the next step  $\lambda$  offspring are sampled from a multi-variate normal distribution,  $\vec{x}_i^{g+1} \sim \mathcal{N}(\langle \vec{x} \rangle^g, (\sigma^g)^2 \mathbf{C}^g)$ ,  $i = 1 \dots \lambda$ , around the mean  $\langle \vec{x} \rangle^g$  of the  $\mu$  parents. After the fitnesses for all  $\lambda$  solutions have been calculated,  $\mu$  best out of  $\lambda$  solutions are recombined to provide a new mean  $\langle \vec{x} \rangle^{g+1}$  for the sampling in the subsequent generation. Besides the mean, the global step-size  $\sigma^g$  and the covariance matrix  $\mathbf{C}^g$  are updated according to Eqs. 2 to 5 of [Hansen and Kern, 2004].

For the optimization of the passenger car design a (2, 12) CMA-ES strategy has been applied using the algorithmic implementation in the Shark machine learning library<sup>3</sup>. The  $k = 16$  and  $k = 12$  parameters of representation  $\mathcal{R}_A$  and  $\mathcal{R}_B$  are coded into the chromosomes of the parent and offspring population. The  $\mu = 2$  individuals of the initial parent population are initialized with the baseline passenger car shape, by means of setting  $\vec{x}^0 = \vec{0}$ . Dependent on the considered parameter variations, the initial step size  $\sigma^0$  has been set to  $\sigma^0 = 0.1$ , and the initial covariance matrix  $\mathbf{C}^{(0)}$  is set to the unity matrix in the first generation. Thus, the  $\lambda = 12$  offspring in the first generation are sampled from a uniform multi-variate normal distribution around  $\vec{x}^0$ . Both optimization runs target the minimization of the overall drag force  $F_D$  constrained by the rear lift  $F_{LR}$ , the volume  $V$  and the maximum control point group displacements. This results in the following fitness function<sup>4</sup>:

$$\begin{aligned}
 f(\vec{x}) &= F_D + \tau_1 \cdot p_1(\vec{x}) + \tau_2 \cdot p_2(V) + \tau_3 \cdot p_3(F_{LR}) \\
 p_1(\vec{x}) &= \sum_{x_i} a, \quad a = \begin{cases} 0 & \text{if } |x_i| \leq 0.3 \\ 1 & \text{if } |x_i| > 0.3 \end{cases} \\
 p_2(V) &= (V - V^c)^2 \\
 p_3(F_{LR}) &= \begin{cases} 0 & \text{if } F_{LR} \leq F_{LR}^c \\ (F_{LR} - F_{LR}^c)^2 & \text{if } F_{LR} > F_{LR}^c \end{cases},
 \end{aligned}$$

where  $p_i$  and  $\tau_i$  define the individual penalty terms and respective weightings. The values for  $\tau_i$  are determined based on experience with  $\tau_1 = 100$ ,  $\tau_2 = 1000$  and  $\tau_3 = 1$ . With  $V^c = 9.40m^3$ , the generated meshes are expected to enclose a similar volume as the initial car. The upper bound for the rear lift  $F_{LR}$  is set to  $F_{LR}^c = 300.00N$ , allowing the rear lift to increase by about

<sup>3</sup><http://shark-project.sourceforge.net>

<sup>4</sup>In the formulation of the quality function and its algorithmic realization, we ignore physical units and implicitly assume that the units of free parameters are chosen accordingly.



## CHAPTER 2. DESIGN SYNTHESIS

---

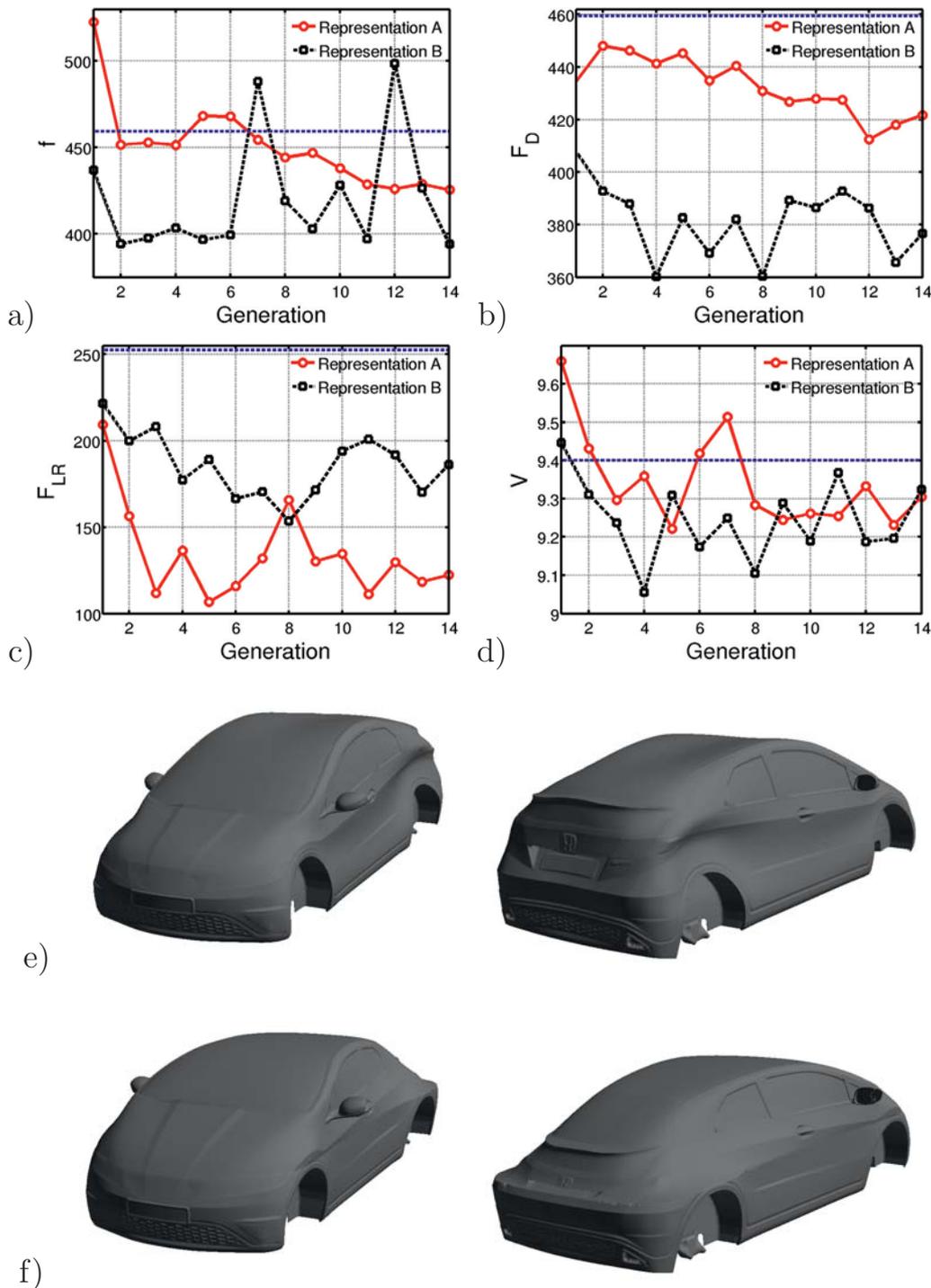
16% compared to the baseline value of  $F_{LR} = 252.53N$ . Furthermore, the search process should keep the control point displacements in a constrained range, punishing extreme deformations.

Two optimization runs have been carried out for 14 generations<sup>5</sup> based on representation  $\mathcal{R}_A$  and  $\mathcal{R}_B$ , respectively. Each optimization run results in 168 different designs. The results of the two runs are summarized in Fig. 2.9. Figs. 2.9 a) to d) visualize the progress of the fitness value, drag force, volume and rear lift force of the best design solution over the different generations, wherein the blue dashed line depicts the performance of the baseline car. Both optimization runs succeeded in developing car shapes that outperform the baseline. In most generations, the best solutions of both runs do not violate any of the volume and rear lift constraints. Especially in early generations, the optimization run based on  $\mathcal{R}_B$  outperforms the optimization run using  $\mathcal{R}_A$  with respect to the fitness and the achieved drag reduction. However, the designs from the run with  $\mathcal{R}_A$  manage to achieve a better performance with respect to the rear lift. The advantage of  $\mathcal{R}_B$  over  $\mathcal{R}_A$  in the optimization results from the more severe constraints used for  $\mathcal{R}_A$ . This is apparent when comparing the shapes of the respective best designs, as depicted in Figs. 2.9 e) and f). As can be seen, the optimization based on  $\mathcal{R}_B$  results in designs with large deformations at the trunk of the car and mesh distortions at the back, ending up in an infeasible car shape.

---

<sup>5</sup>In practice the number of generations is most often limited due to the high computational costs of the fitness evaluation. Improved designs can already be found using a lower number of generations, even the optimizer does not converge.

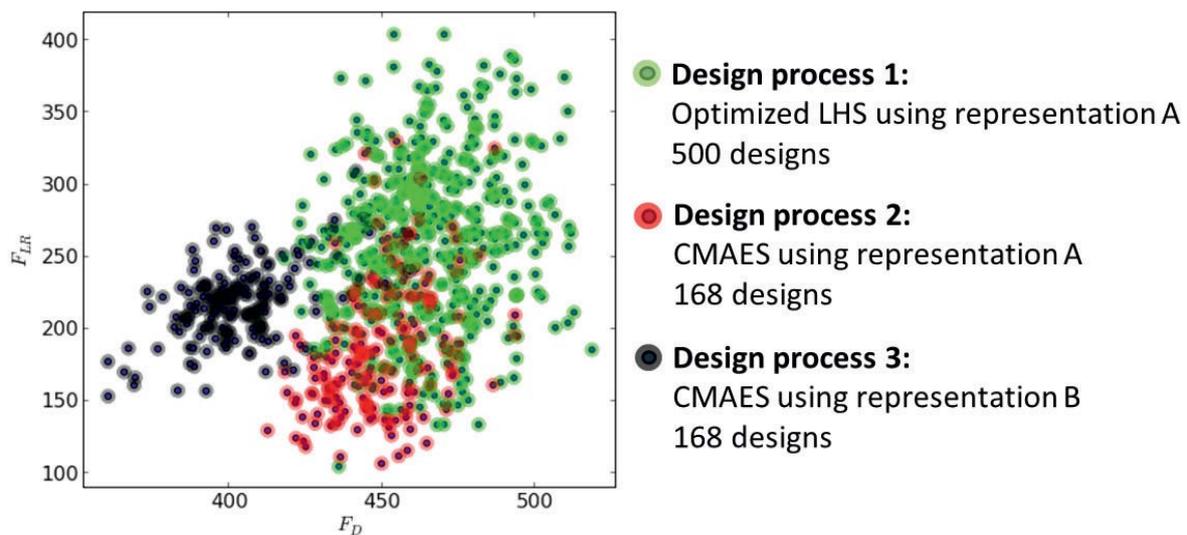
## 2.4. PASSENGER CAR DESIGN SYNTHESIS



**Figure 2.9:** Illustration of the results from two optimizations using a CMA-ES [Hansen and Ostermeier, 2001], based on two design parameterizations A and B. Image a) to d) show the progress of the fitness, drag force, rear lift force and volume, respectively. An isometric view onto the surface of the best performing designs from generation 13 is provided in e) and f).

### 2.5 Interim Summary

In this chapter an introduction to the engineering design synthesis process has been given. The need for a holistic design data analysis in the light of an efficient integrative and simultaneous design process has been motivated, requiring the formulation of a unified design representation. A review of current computer aided engineering technologies revealed a lack in technologies for the analysis of design data beyond individual design processes with well-defined representations and problem settings.



**Figure 2.10:** Summary of all resulting design data, visualized in the performance space with: green showing the results from applying the optimized LHS, red the results from the optimization with CMAES using representation A, and black showing the results from the optimization with CMAES using representation B.

The generation of passenger car geometries and their related performance numbers provides realistic example data for the study of shape mining techniques. The design data has been derived from three distinct processes with different search characteristic and design representations. Fig. 2.10 summarizes the resulting design data. Each car design has been plotted with respect to its performance values, which quantify its respective drag and rear lift force.



# 3

## Meta Design Representation

In the preceding chapter the demand for the definition of a universal design representation as a pre-requisite for a holistic design analysis has been formulated. In this chapter the surface mesh representation is introduced as such a universal meta-design representation. The surface mesh is a discrete representation of the shape of solid geometrical objects. Changes in the surface geometry are the cause for variations in the individual design performance and design characteristics. Those shape variations are determined by local surface differences quantified by the feature difference of corresponding surface points. The analysis of the local surface differences provides first insights into the characteristics of the design process, exemplified based on the passenger car design data at the end of this chapter.

### 3.1 Unstructured Surface Mesh

---

While all surrounding physical objects in the real world are continuous in nature, geometrical objects on the computer are intrinsically discrete. Unstructured surface meshes constitute a discrete geometrical representation of continuous object boundaries on the computer. Constructive methods for retrieving a discrete object representation are under investigation by researchers in the field of discrete geometry. Such methods aim at the reconstruction of objects based on geometrical primitives like points, lines and polygons. For an introduction into discrete geometry the reader is referred to the book of A. M. Bronstein et al. [Bronstein et al., 2008].

Based on the terminology of E. H. Spanier, used in [Spanier, 1966], and the formulations from the work of M. Alexa in [Alexa, 2002], the unstructured surface mesh is defined as follows:

**Unstructured Surface Mesh** An unstructured polygonal surface mesh  $\mathcal{M}$  is a piecewise linear approximation of the boundary of an object. Each surface mesh  $\mathcal{M}$  is defined by a pair  $\mathcal{M} : (\mathcal{V}, \mathcal{K})$ , where  $\mathcal{V}$  is a set of vertices



## CHAPTER 3. META DESIGN REPRESENTATION

---

$\mathcal{V} = (\vec{v}_1, \dots, \vec{v}_n)$ , with  $\vec{v}_i \in \mathbb{R}^m$ , defining the geometric position of  $n$  points sampled from the continuous design boundary. The complex  $\mathcal{K}$  is a set of  $p$  simplices of the form  $\{i_1, i_2, i_3, \dots, i_\mu\}$  with  $i_l, l \in [1..n]$  defining a set of vertices that enclose a polygonal face made up of  $\mu$  segments.

Given a continuous surface  $\mathcal{S}$  the list of vertices  $\mathcal{V}$  build a point cloud by means of a finite set of points in the Euclidean space with  $\mathcal{V} \subseteq \mathcal{S}$ . One of the most important and basic problems in the construction of surface mesh representations is the *sampling* of the continuous surface. The choice of the sampling technique is crucial and decides about the amount of information lost when transferring the object into a discrete representation. The information that has to be retained in the mesh depends apparently on the task at hand. CAD programs allow to constraint the sampling by two parameters that define the quality of the final mesh. The *sag* value controls the approximation quality of curved parts of the surface, while the *step* parameter constraints the maximum length between neighboring vertices.

The polygonal faces, defined by the simplices  $p \in \mathcal{K}$ , make up a list of surface patches building a local linear approximation of the surface. Derived from the normal vectors of the surface patches, the normal vector  $\vec{n}_i \in \mathbb{R}^3$  at the position of each vertex  $\vec{v}_i$  can be derived, resulting in the vertex list  $\mathcal{N} = (\vec{n}_1, \dots, \vec{n}_n)$ ,  $\vec{n}_i \in \mathbb{R}^3$ . The normal vector  $\vec{n}_i$  has a defined direction perpendicular to the surface mesh and provides local gradient information at the position of vertex  $\vec{v}_i$ .

In the aerodynamic design process discrete surface meshes are mainly adopted as input for rapid prototyping machines. Therefore, 3D CAD objects are tessellated and transferred to the STL (Surface Tessellation Language) format, which has become a standard for rapid prototyping machines. Typically, triangular facets are used to approximate the surface. Alongside rapid prototyping processes, discrete surface mesh representations gain in importance for a huge variety of applications. As an example, automatic CFD mesh generation tools process upon discrete mesh representations [Aftosmis et al., 1999]. The simplicity of surface meshes and the fact that nearly all design representation can be transferred into this geometrical representation makes it an adequate choice to use it as the basic representation for sharing information about designs between various computational applications and design teams.

In this study discrete surface meshes are adopted as a universal representation for analyzing stocks of design data. The representation covers all necessary information essential to analyze the shape of solid objects. The unstructured surface mesh as a general representation allows the analysis of local shape variations and their influence on the functional properties con-



cerning a set of designs. The surface mesh is independent of the parametric representation that has been used during the design process. However, it has to be noted that surface meshes solely represent the object surface and leave an explicit modeling of the internal structure of an object aside.

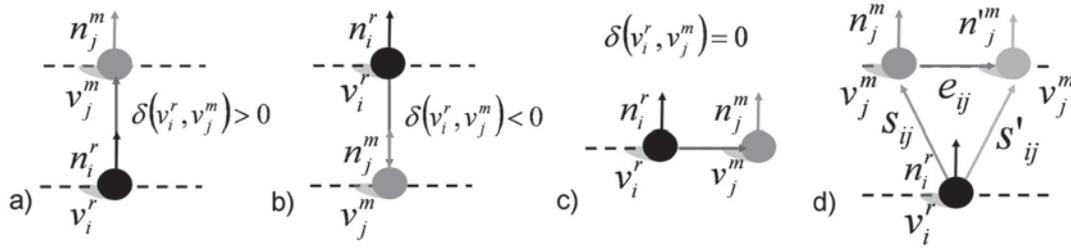
## 3.2 Local Surface Differences

---

Each surface mesh is defined with respect to a three dimensional global coordinate system. The origin of the coordinate system can, and seems often be arbitrarily chosen. In most applications the absolute position of the surface is irrelevant. Rather, the difference between surface features provides valuable insights for the cause of the change in functional design properties. For example considering aerodynamic properties of a design, the performance of an object exposed to airflow is obviously independent of the position of the origin, as long as the relative position to the source of the incoming flow and the relative position to surrounding objects do not change. The quantification of the *local surface difference* is intended to quantify the characteristics that are potentially the cause of functional changes. Related to unstructured surface meshes the local surface difference is formally defined as follows.

**Local Surface Difference** Given two surface mesh representations  $\mathcal{M}_r$  and  $\mathcal{M}_m$  with a finite set of vertices  $\mathcal{V}_r$  and  $\mathcal{V}_m$  respectively, the local surface difference  $\Delta_{i,j}^{r,m} = D(f_i^r, f_j^m)$  is defined for each vertex  $\vec{v}_i^r$  as the disparity between the feature  $f_i^r$  assigned to  $\vec{v}_i^r$  and the feature  $f_j^m$  of its corresponding vertex  $v_j^m$ .

The calculation of the local surface differences involves the quantification of local surface features at the position of each vertex, the identification of its corresponding vertex and the final calculation of the disparity between the individual local features. In the remainder of this section the calculation of the displacement and the curvature disparity as specific measures for local surface difference are introduced, followed by a description of methods for identifying corresponding vertices. The change of vertex positions in form of the displacement and the change in curvature are expected to be relevant especially for investigating data from aerodynamic applications. However, the selected features are strongly dependent on the task at hand and the functional characteristics under investigation. The selection of the features should be constantly reviewed and adjusted to the given targets.



**Figure 3.1:** Examples of the displacement measure. Figures a) and b) illustrate that a vertex displacement parallel (anti-parallel) to the normal direction results in a positive (negative) displacement value. A displacement perpendicular to the normal vector results in a displacement value of zero, as shown in c). Figure d) illustrates the error when calculating the displacement measure, which results from the discretization of the surface and the error when estimating corresponding points.

### 3.2.1 Vertex Displacement

The displacement measure describes the position of a vertex with respect to another design. One way to capture this information is to use the difference vector  $\vec{s}_{ij} = \vec{v}_i^r - \vec{v}_j^m$ , which is the difference between vertex  $\vec{v}_i^r$  of mesh  $\mathcal{M}_r$  and the corresponding vertex  $\vec{v}_j^m$  of mesh  $\mathcal{M}_m$ . The difference vector captures the exact information about the displacement between  $\vec{v}_i^r$  and  $\vec{v}_j^m$ . However, the difference vector is sensitive to possible errors resulting from wrong estimations of the corresponding vertices or from different sampling methods of the surfaces of the geometries. Furthermore, the difference vector requires  $d = 3$  parameters to describe the displacement of a vertex in  $\mathbb{R}^3$ . Thus, to capture the displacement between two complete surface meshes the number of parameters is  $3 \cdot n_r$ , where  $n_r$  equals the number of vertices that belong to the reference mesh. For a reduced representation of local surface differences, the following displacement measure is suggested instead:

$$\Delta_{i,j}^{r,m} \triangleq \delta_{i,j}^{r,m} = \delta(\vec{v}_i^r, \vec{v}_j^m) = (\vec{v}_j^m - \vec{v}_i^r) \circ \vec{n}_i^r, \delta \in (-\infty, +\infty) \quad (3.1)$$

The displacement measure is defined as the projection of the difference vector  $\vec{s}_{ij} = (\vec{v}_i^r - \vec{v}_j^m)$  onto the normal vector  $\vec{n}_i^r$  of vertex  $\vec{v}_i^r$  of the reference design  $\mathcal{M}_r$ . According to [Taubin, 1995], the vertex normal can be estimated from the adjacent triangular faces as a weighted sum of the face normals, where the weights are proportional to the area of each facet. The absolute value of the displacement provides information on the amount, and the sign of the displacement measure in conjunction with the vertex normal provides information on the direction of the vertex modification. The normal vector  $\vec{n}_i^r$  points towards the normal or positive direction of vertex modification.

## 3.2. LOCAL SURFACE DIFFERENCES

The displacement measure is by definition a non-symmetric vector quantity containing both the magnitude and the direction of vertex modification. If the modified vertex lies above the tangential plane described by the normal vector of the reference vertex, the displacement measure is positive, see Fig. 3.1 a). Whereas, if the vertex lies below the tangential plane, Fig. 3.1 b), the displacement measure is negative. In the special case, where the modified vertex is located within the tangential plane, the displacement measure is zero as shown in Fig. 3.1 c), which makes sense because we have to assume that the difference is a result of a different triangulation. If the reference vertex has been modified along the line described by the normal vector, the amount of the displacement measure equals the Euclidean distance between the reference and the modified vertex.

As Fig. 3.1 d) indicates, the displacement value contains an error, which is mainly caused by the discretization of the surface using triangulation and due to the correspondence problem. Formally, this can be written as,

$$\delta_{i,j}^{r,m} = (\vec{s}_{ij} + \vec{e}_{ij}) \circ \vec{n}_i^r = \vec{s}_{ij} \circ \vec{n}_i^r + \vec{e}_{ij} \circ \vec{n}_i^r, \quad (3.2)$$

where  $\vec{e}_{ij}$  describes the error between the ideal displacement value and the measured displacement value. Under the assumption that the curvature of both surfaces  $\mathcal{M}_r$  and  $\mathcal{M}_m$  is similar at the position of the corresponding vertices it follows that  $\vec{n}_i^r \approx \vec{n}_j^m$ . Then, the error term from equation (3.2) simplify as:

$$\vec{e}_{ij} \circ \vec{n}_i^r \approx \vec{e}_{ij} \circ \vec{n}_j^m. \quad (3.3)$$

With  $\vec{e}_{ij} \circ \vec{n}_j^m = |\vec{e}_{ij}| \cos(\angle(\vec{e}_{ij}, \vec{n}_j^m))$ , if additionally a smooth surface or a small error  $|\vec{e}_{ij}|$  is assumed,  $\vec{e}_{ij}$  is perpendicular to  $\vec{n}_j^m$  and hence  $\cos(\angle(\vec{e}_{ij}, \vec{n}_j^m)) \approx 0$ . Thus the error term becomes zero. Therefore, the displacement measure is relatively insensitive to small errors arising from the surface triangulation or from an incorrect estimation of corresponding points.

Another advantage of the displacement measure compared to the difference vector is that only  $n_r$  parameters are required to describe the differences between two unstructured surface meshes, where  $n_r$  equals the number of vertices of the reference mesh.

The displacement measure is not invariant to rotation and translations of the geometric object. Thus, a translation of the entire object can result in large displacement values but might have no difference in the functional characteristics of an object. Therefore, in some scenarios the normalization of the surface mesh can be required a priori to the calculation of the displacements.



### 3.2.2 Curvature Disparity

Slope and curvature are invariant features of the surface as a two dimensional manifold, intensively studied in the field of differential geometry, analyzing differential and integral properties of geometric objects [Botsch et al., 2010; Bronstein et al., 2008]. Differential properties characterize transformation and rotation invariant properties, related to the mathematical definition of the first and second fundamental form, related to the first and second partial derivation of the surface. The two fundamental forms build up a mathematical basis for estimating local properties of the surface like elongation, angle, area or curvature.

For the definition of the curvature, lets first consider the continuous description of a 2D surface  $S \in \mathbb{R}^3$ . Given the parametric form  $\vec{s}(u, v)$  of  $S$ , the first partial derivatives with respect to  $u$  and  $v$  are  $\vec{s}_u = \vec{s}_u(u, v)$  and  $\vec{s}_v = \vec{s}_v(u, v)$  respectively. Further given a point  $\vec{p} \in S$ , with  $\vec{p} = \vec{s}(0)$ , the vectors  $\vec{s}_u$  and  $\vec{s}_v$  define a tangential plane at the point  $\vec{p}$ . The unit vector perpendicular to the tangential plane is the result of the normalized cross product,

$$\vec{n} = \frac{\vec{s}_u \times \vec{s}_v}{|\vec{s}_u \times \vec{s}_v|}. \quad (3.4)$$

The orientation of  $\vec{n}$  reflects the slope at the position of  $\vec{p}$ . according to [Botsch et al., 2010], given the partial derivatives  $\vec{s}_u$ ,  $\vec{s}_v$  and the normal vector  $\vec{n}$ , the first  $I$  and second fundamental  $II$  form are defined as:

$$I = \begin{bmatrix} E & F \\ F & G \end{bmatrix} := \begin{bmatrix} \vec{s}_u^T \vec{s}_u & \vec{s}_u^T \vec{s}_v \\ \vec{s}_u^T \vec{s}_v & \vec{s}_v^T \vec{s}_v \end{bmatrix}, \quad (3.5)$$

$$II = \begin{bmatrix} e & f \\ f & g \end{bmatrix} := \begin{bmatrix} \vec{s}_{uu}^T \vec{n} & \vec{s}_{uv}^T \vec{n} \\ \vec{s}_{uv}^T \vec{n} & \vec{s}_{vv}^T \vec{n} \end{bmatrix}. \quad (3.6)$$

For each point  $\vec{p}$  on  $S$ , an infinite number of curves  $C : \vec{c}(t) = \vec{s}(u(t), v(t)) \in S$  through  $\vec{p}$  exist, where each curve is the result of a plane that is cutting the surface in the direction of the normal vector  $\vec{n}$  and an arbitrary tangent  $\vec{t}$ , lying in the tangential plane of  $\vec{p}$ . While the first derivative of  $C$  at  $\vec{p}$  defines the tangent vector  $\vec{t}$ , the second derivative defines the curvature vector, denoted as  $\kappa \vec{n}_c$ , with  $\vec{n}_c$  being the normal vector to the curve at  $\vec{p}$ . The curve normal vector is perpendicular to  $\vec{t}$  and  $\kappa$  is a nonnegative scalar value. With  $\phi$  defining the angle between the surface normal  $n$  and the curve normal  $n_c$ , the *normal curvature* in the direction of the tangent  $\vec{t}$  is defined as:



## 3.2. LOCAL SURFACE DIFFERENCES

$$\kappa_n(\vec{t}) = \kappa \cos \phi. \quad (3.7)$$

All curves  $C$  that have the same tangent vector at  $\vec{p}$ , have the same normal curvature  $\kappa_n$  [Hameiri and Shimshoni, 2003]. The normal curvature can as well be formulated in terms of the first and second fundamental form:

$$\kappa_n(\vec{t}) = \frac{\vec{t}^T I I \vec{t}}{\vec{t}^T I \vec{t}}. \quad (3.8)$$

The normal curvature can take positive and negative values. Apparently, as can be seen from equation (3.7), the sign of the curvature depends on the orientation of the surface normal  $\vec{n}$ . Among all possible directions of  $\vec{t}$ , the maximum and minimum curvatures, denoted as  $k_1$  and  $k_2$  respectively, are called the *principal curvatures*. The related directions define the *principal directions*,  $\vec{t}_1$  and  $\vec{t}_2$ . The principal directions build together with the surface normal vector an orthogonal system, the Darboux frame [Bronstein et al., 2008], where each curvature can be expressed as a weighted combination of the principal curvatures.

From the principal curvatures a number of quantities can be calculated that condense the curvature information related to one point on the surface into a scalar value. The *mean curvature* builds the arithmetic mean between the principal curvatures:

$$H = \frac{\kappa_1 + \kappa_2}{2} = \frac{1}{2\phi} \int_0^{2\pi} \kappa_n(\phi) d\phi, \quad (3.9)$$

what equals the integral over the normal curvatures in any tangential direction. The mean curvature  $H$  quantifies the flatness of the surface at a point  $\vec{p}$ , see [Chen-shi and Guo-zhao, 2005]. While  $H$  is defined as the arithmetic mean, the *Gaussian curvature* is the product of the principal curvatures:

$$K = \kappa_1 \cdot \kappa_2. \quad (3.10)$$

The Gaussian curvature  $K$  becomes negative in case that only one of the principal curvatures is negative, and such highlights surface curvatures that are present in hyperbolic patches. The Gauss curvature allows to classify surface patches into one of the following categories, elliptic with  $K > 0$ , parabolic with  $K = 0$  and hyperbolic with  $K < 0$ .

Considering discrete polygonal surface meshes, the target is to estimate the principal curvatures at the position of each vertex  $\vec{v}_i$ , given the adjacent polygonal faces and vertices [Taubin, 1995]. Different approaches exist to estimate the curvatures based on sample points. Those algorithms are categorized by Rusinkiewicz, in [Rusinkiewicz, 2004].

The first group of algorithms approach the curvature estimation by fitting an analytical surface through neighboring vertices and calculate the curvature analytically, as described above. Rusinkiewicz refers to those algorithms as *patch fitting methods*. The related methods mostly differ in the way the surface patch is fitted to the vertices. Examples are given in [Besl and Jain, 1986; Flynn and Jain, 1989].

A more direct approach attempts the estimation of the principal curvatures, by calculating the normal curvatures in the direction of the neighboring vertices and calculating the Eigenvalues of the resulting matrix [Taubin, 1995; Hameiri and Shimshoni, 2003], referred to as *normal curvature-based methods*. An alternative approach to the normal curvature-based estimation of principal curvatures is the direct approximation of the curvature tensor using *tensor averaging methods* [Rusinkiewicz, 2004].

Finally, *the curvature disparity* defines the curvature difference between the estimated curvature at the position of vertex  $\vec{v}_i^r$  of mesh  $\mathcal{M}_r$  and the curvature estimated at the position of vertex  $\vec{v}_j^m$  of mesh  $\mathcal{M}_m$ , with

$$\Delta_{i,j}^{r,m} \triangleq \Delta H_{i,j} = H_j - H_i, \quad (3.11)$$

where  $H_i$  and  $H_j$  define the estimated mean curvature related to vertex  $\vec{v}_i^r$  and  $\vec{v}_j^m$ , respectively. Similarly, the Gauss curvature disparity is defined as,

$$\Delta_{i,j}^{r,m} \triangleq \Delta K_{i,j} = K_j - K_i, \quad (3.12)$$

where  $K_i$  and  $K_j$  define the Gauss curvature estimated at the position of  $\vec{v}_i^m$  and  $\vec{v}_j^m$ . The measures of mean and the Gauss curvature disparity define alternative measures of local surface variations, which are translation and rotation invariant.

### 3.2.3 Identification of Corresponding Vertices

An appropriate identification of corresponding vertices is essential for measuring the correct feature difference between vertices of two different surfaces. Wrong estimates will lead to an error in the measurements and hence to errors in any subsequent knowledge extraction step.

Let  $\mathcal{M}_r$  and  $\mathcal{M}_m$  be two unstructured surface meshes, where  $\mathcal{M}_r$  is a baseline or reference mesh and  $\mathcal{M}_m$  is the target or modified mesh. The main objective in solving the correspondence problem is to find an appropriate function  $f$  which maps each vertex  $\vec{v}_i^r \in \mathcal{V}_r$  from the reference mesh  $\mathcal{M}_r$  to a corresponding vertex  $\vec{v}_j^m \in \mathcal{V}_m$  of the target mesh  $\mathcal{M}_m$ . More formally this is written as:

### 3.2. LOCAL SURFACE DIFFERENCES

$$f : \mathcal{V}_r \rightarrow \mathcal{V}_m, \quad \mathcal{V}_r \in \mathcal{M}_r, \quad \mathcal{V}_m \in \mathcal{M}_m \quad (3.13)$$

Thus, for each vertex  $i$  a corresponding vertex  $j$  can be found that minimizes  $f$ , implementing a local matching strategy.

A global solution to the correspondence problem is ambiguous, and a universal transfer function  $f$  that leads to an "exact" mapping without providing any prior knowledge about the design can not exist. A correct identification of the corresponding points is solely possible if the transformation that maps  $\mathcal{M}_r$  to  $\mathcal{M}_m$  is known. Constraint by the diversity of possible design transformations, a specific mapping function or algorithm has to be chosen which is tuned to the considered design data characteristics. In this work, the assumption is made that no structural design modifications are applied to any of the designs. Only local deformations are considered, so that the global structure of the object remains widely unchanged.

Taking into account solely the vertex location, the Euclidean distance between vertices can be used to find a solution for  $f$ ,

$$f(\vec{v}_i^r) = \min_{j \in n_m} \{ |\vec{v}_i^r - \vec{v}_j^m| \}, \quad (3.14)$$

where  $n_m$  equals the number of vertices that make up the surface mesh of the target design, and where  $|\vec{v}_i^r - \vec{v}_j^m|$  defines the Euclidean distance between pairs of vertices. In general, it is possible that one vertex from mesh  $\mathcal{M}_m$  is assigned to more than one vertex of mesh  $\mathcal{M}_r$ .

The identification of corresponding vertices using the Euclidean distance has its weaknesses in areas where edges or solid structures are deformed. In such areas, adding information about the local curvature to the functional  $f$  can overcome otherwise false matches. The following objective function uses normal information to improve the identification of corresponding vertices,

$$f_n(\vec{v}_i^r) = \min_{j \in n_m} \{ |\vec{v}_i^r - \vec{v}_j^m| \cdot (2 - \vec{n}_i^r \circ \vec{n}_j^m) \}, \quad (3.15)$$

where  $|\vec{v}_i^r - \vec{v}_j^m|$  again defines the Euclidean distance and  $\vec{n}_i^r \circ \vec{n}_j^m$  measures the difference between the normal vectors of pairs of vertices. The objective function has been defined in a way that vertex  $\vec{v}_j^m$  is assigned as corresponding vertex, which is closest to  $\vec{v}_i^r$ , and which has the most similar normal vector. On the one hand if the normal vectors have the same orientation the scalar product equals one and hence the function relies completely on the spatial distance between vertices. If on the other hand the normal vectors are perpendicular (pointing in opposite direction) the objective function is two times (three times) the Euclidean distance. The objective function is a simplification of the one used by [Wang et al., 2000]. Wang additional



## CHAPTER 3. META DESIGN REPRESENTATION

---

added information about the Gauss curvature to the objective function  $f$ . Following this idea,  $f_n$  or  $f$  can be enhanced by adding information about the curvature disparity between vertices:

$$f_H(\vec{v}_i^r) = \min_{j \in n_m} \{ |\vec{v}_i^r - \vec{v}_j^m| \cdot (2 - \vec{n}_i^r \circ \vec{n}_j^m) + w_H \cdot \Delta H_{i,j}^* \}, \quad (3.16)$$

or

$$f_K(\vec{v}_i^r) = \min_{j \in n_m} \{ |\vec{v}_i^r - \vec{v}_j^m| \cdot (2 - \vec{n}_i^r \circ \vec{n}_j^m) + w_H \cdot \Delta K_{i,j}^* \}, \quad (3.17)$$

where  $w_H$  and  $w_K$  define the weights that specify the influence of the curvature on the objective function. Although the functionals  $f_n$ ,  $f_H$  and  $f_K$  are expected to improve the identification of corresponding vertices compared to  $f$ , the correct trade-off or weighting (e.g.,  $w_H$  and  $w_K$ ) between the Euclidean distance and the normal or curvature information is hard to define a priori and needs to be specified empirically, dependent on the design variations considered.

In [Wang et al., 2000], the authors further investigated the use of the geodesic distance to improve the identification of corresponding vertices. If affine transformations like rotation, scaling or translation are applied to the designs, enhanced algorithms, as suggested by [Besl and McKay, 1992; Feldmar and Ayache, 1996; Jost, 2002], need to be adopted. In the iterative closest point algorithms a global transformation is searched that maps one design into the other while minimizing the distance between vertices. In [Laskov and Kambhamettu, 2001], Laskov provides a comparison of five different iterative closest point algorithms. Other authors like Chui and Rangarajan [Chui and Rangarajan, 2000] even suggest applying non-rigid local deformations to match both designs before identifying the corresponding points. Designs are deformed to simplify the subsequent search for corresponding points. However, applying deformation methods before identifying corresponding vertices is computational expensive.

**Examples** The following examples target to illustrate the coverage and limitations of the correspondence estimation using the Euclidean and normal weighted distance, referring to equation (3.14) and (3.15), respectively. The assignment of corresponding points is exemplified based on three different classes of 2D shape variations. The considered reference and modified shapes are depicted in Figs. 3.2, within the first two rows of the illustrations. The respective reference and modified designs are sampled along the object shape using a limited set of 2D points. Additionally, the normal vectors are calculated at the position of each vertex, pointing towards the outside of the shape.

## 3.2. LOCAL SURFACE DIFFERENCES

---

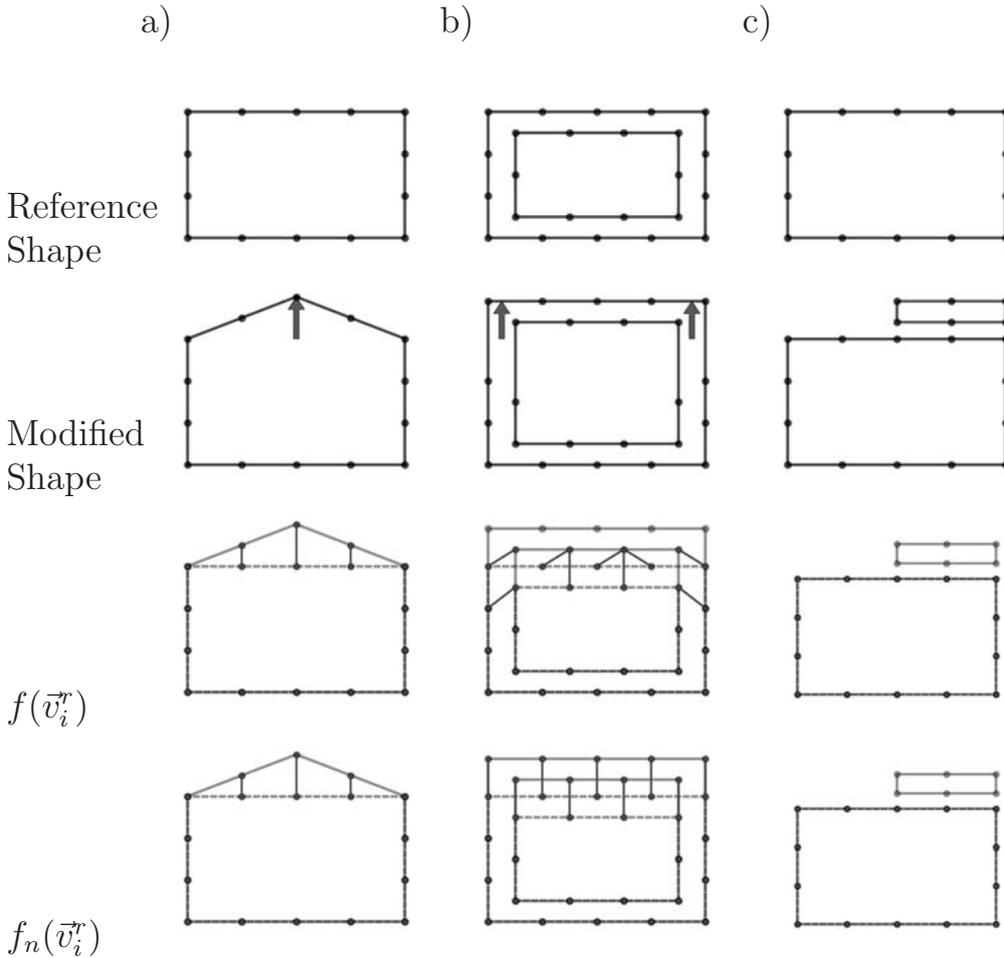
As highlighted in Figs. 3.2, certain points of the shape are manipulated to result in the modified shapes, representing three classes of object manipulations referring to *small a)*, *complex b)* and *structural modifications c)*.

Minor modifications comprise small deformations of closed object shapes, where deformations are small with respect to the overall shape size. In a second example, major modifications are applied to a non-closed object shape, exemplifying modifications of more complex objects, e.g., realizing modifications of a solid bounded object. Structural modifications are to be considered if new sub-components of a shape are added or removed from the reference shape, e.g., when adding a rear wing to a car.

The corresponding points are estimated based on the reference shape with respect to the modified shape, implementing a uni-directed mapping between the set of points. The third and fourth row of the illustrations in Fig. 3.2 show the results of the mapping using the Euclidean and normal weighted distance in an exhaustive search attempt. Each point of the reference shape is assigned to its corresponding point on the modified shape. The different graphs depict the contour of the reference shape (dashed line), the modified shape (solid line) and the pairs of corresponding points connected by a solid red line. From the results, it can be seen that the Euclidean and the normal weighted search strategy perform equally well, regarding small deformations. With the knowledge about the applied deformation, all points of the reference shape are linked to the correct corresponding points of the modified shape.

However, if more complex deformations are applied, as depicted in example b), the exhaustive search purely based on the Euclidean distance fails in resolving the correct correspondence. Points are directly assigned to their closest analogon, without considering, e.g., the inside and outside of an object. This can be resolved by considering normal information in the target function, as shown in example b). Hence, using the normal weighted distance can resolve the correct correspondence even for major shape modifications.

As depicted in Fig. 3.2 c), if structural modifications are applied to the reference shape, both strategies fail in assigning corresponding vertices correctly. In such cases, a correct assignment can only be resolved by applying more enhanced search strategies. Different assumptions about possible shape variations need to be considered, e.g., implementing a bi-directional mapping or other heuristics to resolve possible conflicts. Even sophisticated techniques based on the iterative closest point matching approach [Besl and McKay, 1992; Feldmar and Ayache, 1996] would not be able to solve the correspondence problem for such a class of shape variations.



**Figure 3.2:** The graphics in the first and second row illustrate three different reference and modified 2D shapes, representing three simple examples for a) small, b) complex and c) structural object deformations. Given individual sets of sampling points along the object shapes, the graphics in the third and fourth row illustrate the results from the identification of corresponding points using either of the two objective functions  $f(\vec{v}_i^r)$  or  $f_n(\vec{v}_i^r)$ , where the reference shape is visualized with dashed and the modified shape with solid lines. Red solid lines visualize the estimated correspondences.

### 3.3 Data Generation

Design and optimization processes result in a set of various modified designs. For each design the surface mesh shape representation and the performance indicators are derived. The shape and performance data are collected in a design database. By comparing individual design surfaces, the surface features and feature disparities are calculated according to the process flow in Fig. 3.3.

In the first step, a reference design, sometimes referred to as baseline or pivot design, needs to be defined. The related surface representation  $\mathcal{M}^r$  is retrieved from the design database. The choice of the reference design plays an important role. The feature disparities are all calculated with respect to the chosen reference design. Thus, all subsequent information and knowledge extraction steps need to be seen in the light of the reference design. Which design or which designs are chosen as reference design depends on the question asked upon the design domain. From  $\mathcal{M}^r$  the feature vector  $\mathbf{f}^r$  is evaluated and temporarily stored for the data generation process.

In the next step, a limited set of designs and their corresponding surface representations  $\mathcal{M}^m$  are chosen. Let  $N_D$  be the number of selected designs, in an iterative process each out of  $N_D$  designs is processed, features are calculated, corresponding vertices are determined and the feature disparities  $\Delta^{r,m}$ , as well as the performance differences  $\phi^{r,m}$  are calculated. While  $\Delta^{r,m}$  describes local differences between design surfaces,  $\phi^{r,m}$  quantifies the difference mostly with respect to global functional properties of a design. For each design  $m$  the length of  $\Delta^{r,m}$  and  $\phi^{r,m}$  are the same. Hence, for each vertex  $i$  of the reference design a disparity vector can be formulated,

$$\Delta^r = (\Delta_1^r \Delta_2^r \dots \Delta_n^r), \quad (3.18)$$

resulting in a disparity matrix:

$$\Delta^r = \begin{bmatrix} \Delta_1^{r,1} & \Delta_2^{r,1} & \dots & \Delta_n^{r,1} \\ \Delta_1^{r,2} & \Delta_2^{r,2} & \dots & \Delta_n^{r,2} \\ \dots & \dots & \dots & \dots \\ \Delta_1^{r,N_D} & \Delta_2^{r,N_D} & \dots & \Delta_n^{r,N_D} \end{bmatrix}, \quad (3.19)$$

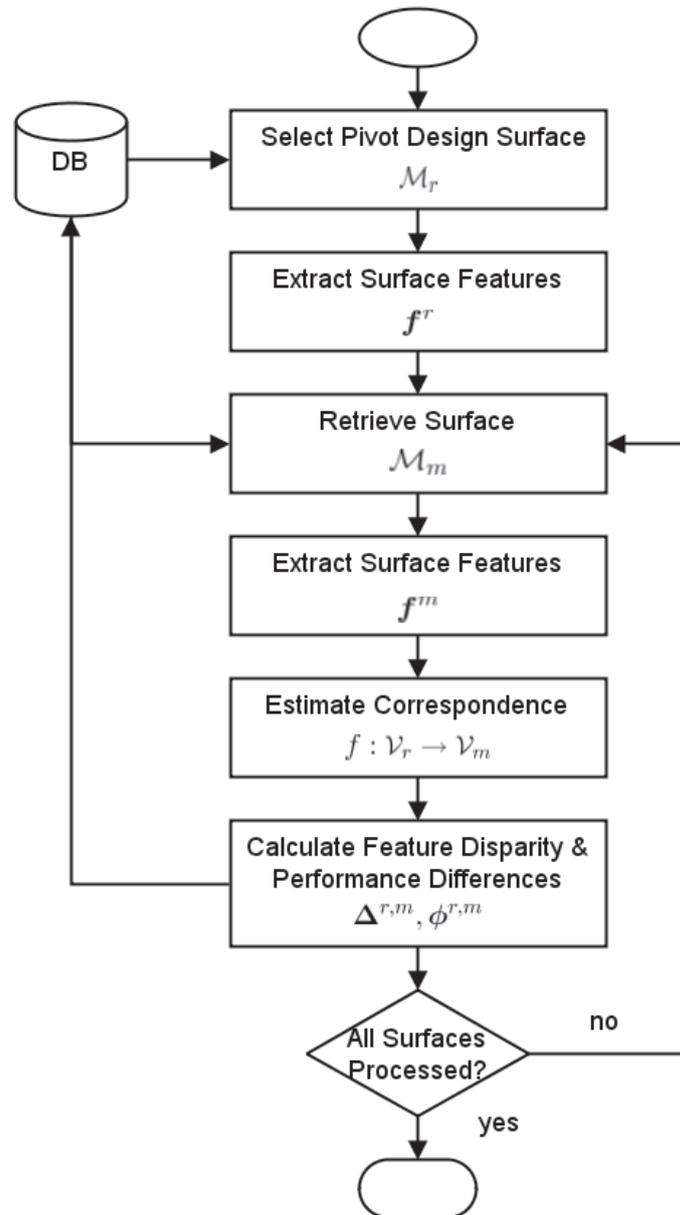
which comprises all local differences with respect to the chosen reference mesh  $\mathcal{M}^r$ . The number of vertices  $n$  can vary between different designs or surfaces meshes. Hence, the number of features contained in the resulting data set depends as well on the reference design chosen.

In the same way, for each performance measure a difference vector is formalized,

$$\phi^r = \{\phi_1^r, \phi_2^r, \dots, \phi_l^r\}, \quad (3.20)$$

and resulting in the performance difference matrix, formally defined as follows,

$$\phi^r = \begin{bmatrix} \phi_1^{r,1} & \phi_2^{r,1} & \dots & \phi_m^{r,1} \\ \phi_1^{r,2} & \phi_2^{r,2} & \dots & \phi_m^{r,2} \\ \dots & \dots & \dots & \dots \\ \phi_1^{r,N_D} & \phi_2^{r,N_D} & \dots & \phi_m^{r,N_D} \end{bmatrix}, \quad (3.21)$$



**Figure 3.3:** Flow chart summarizing the process for generating feature disparity and performance difference data with respect to a chosen reference design.

comprising all information about the performance differences between designs.

Finally, the resulting dataset is build out of two data matrices,  $\Delta^r$  and  $\phi^r$ . This dataset defines the basis for the analysis of the design data and for all subsequent steps of the shape mining process.

## 3.4 Quantification and Analysis of Design Variations

---

Analyzing local modifications in form of vertex displacement or curvature disparity can provide engineers with relevant insights about the exploration of the design space. Based on basic statistics, two measures are introduced: the *relative mean vertex disparity*, which provides information on how the characteristic of a vertex changed with respect to a chosen reference design, and the *relative design variance*, which identifies the vertices that have been modified most frequently. This basic statistical information also allows to identify those design modifications, which have not been considered in the design and optimization process so far.

### 3.4.1 Relative Mean Vertex Disparity

The arithmetic mean or sample mean is a basic statistical measure calculating the center of a given dataset. Concerning the design variation data from equation (3.19), the calculation of the mean is related to the average variation of a set of designs with respect to the chosen reference design. In an design or optimization process it provides information about the tendency of this process with respect to design changes. Given the disparity matrix as specified in equation (3.19), the *relative mean vertex disparity* is defined as:

$$E(\Delta_i^r) = \bar{\Delta}_i^r = \frac{1}{N_D - 1} \sum_{m=1, m \neq r}^{N_D} \Delta_{i,j}^{r,m}, \quad (3.22)$$

where  $\Delta_{i,j}^{r,m}$  is the quantified disparity between vertex  $v_i^r$  of the reference design  $r$  and its corresponding vertex  $v_j^m$  of the design  $m$  from the dataset.  $E(\Delta_i^r)$  quantifies the expected disparity of a selected vertex  $v_i^r$  among the considered design dataset. It has to be noted that the integration of the mean vertex disparity for different reference designs might vanish, depending on the calculation of the disparity between vertex features. For a skewed distribution of the vertex disparities the calculation of the median should be preferred over the mean value, see [Kantardzic, 2001].

Depending on the number of vertices  $n$  of the reference design  $\mathcal{M}_r$  and the size of the considered design dataset, the computation of the relative mean vertex disparity can get computational expensive. In an online configuration, where new designs are added successively, the update rule for calculating the mean value is applicable:

$$E(\Delta_i^r)' = \frac{1}{N_D} ((N_D - 1) \cdot E(\Delta_i^r) + \Delta_{i,j}^{r,N_D+1}), \quad (3.23)$$

where  $\Delta_{i,j}^{r,N_D+1}$  is the vertex disparity between vertex  $v_i^r$  and its corresponding vertex  $v_i^{(N_D+1)}$  of the newly added surface mesh  $\mathcal{M}_{N_D+1}$ . The use of the update rule requires to permanently store the calculated mean values.

### 3.4.2 Relative Design Variance

In statistics the dispersion of the given parameter values from the arithmetic mean is typically evaluated using the *variance* or *standard deviation*. Measures of dispersion quantify the spread from the mean of the data values and should be applied only when the mean is used for calculating the center of the data values [Kantardzic, 2001]. The calculation of the *relative design variance* or *standard deviation* defines the expected variation from the mean disparity value within the considered design dataset. Given the disparity matrix from equation (3.19), calculated with respect to a particular reference mesh  $\mathcal{M}_r$ , the relative design standard deviation is formally defined as:

$$\sigma_i^r = \sqrt{\frac{1}{N_D - 1} \sum_{m=1, m \neq r}^{N_D} (\Delta_{i,j}^{r,m} - \bar{\Delta}_i^r)^2}, \quad (3.24)$$

with  $\bar{\Delta}_i^r$  defining the relative mean disparity as defined in equation (3.22). According to the definition of the relative design standard deviation, the relative design variance is defined as  $Var(\Delta_i^r) = (\sigma_i^r)^2$ .

An online update of the relative design variance can be derived by adopting the displacement law from statistics. Accordingly, the update rule of the variance is calculated as follows,

$$Var(\Delta_i^r)' = Var(\Delta_i^r) + (E(\Delta_i^r))^2 - (E(\Delta_i^r)')^2, \quad (3.25)$$

where  $E(\Delta_i^r)'$  is calculated according to equation (3.23). For an online update of the variance, the previous mean,  $E(\Delta_i^r)$ , and variance,  $Var(\Delta_i^r)$ , values have to be permanently stored.

The *overall design variation* is defined as the expected standard deviation for all possible reference designs, formalized as:

$$\sigma_{\Delta_i} = \sqrt{\frac{1}{N_D(N_D - 1)} \sum_{r=1}^{N_D} \sum_{m=1, m \neq r}^{N_D} (\Delta_{i,j}^{r,m} - \bar{\delta}_i^r)^2}. \quad (3.26)$$

### 3.5. APPLICATION TO PASSENGER CAR DESIGN

---

As noted in Section 3.4.1, the sum of the mean vertex disparity over multiple reference designs might result in a vanishing mean disparity value. In such a case the overall design variation can be approximated as:

$$\sigma_{\Delta_i} \approx \sqrt{\frac{2}{N_D(N_D - 1)} \cdot \sum_{r=1}^{N_D} \sum_{m=r+1}^{N_D} (\Delta_{i,j}^{r,m})^2}. \quad (3.27)$$

## 3.5 Application to Passenger Car Design

---

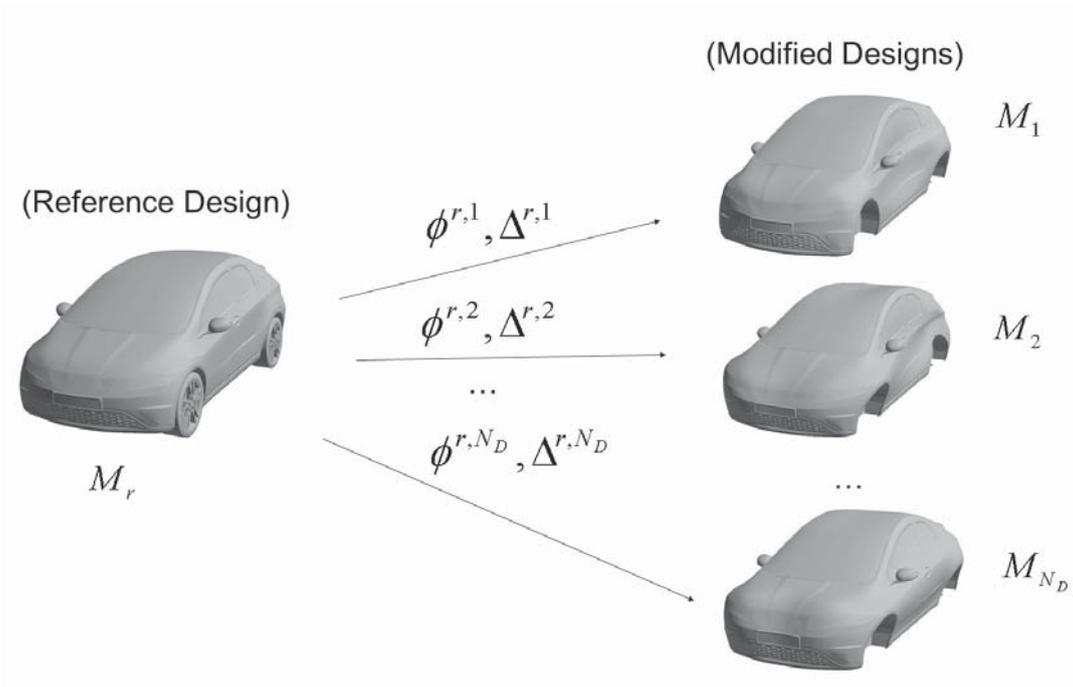
The design data from the three design processes, described in Chapter 2, build the basis for the evaluation of the shape disparities. For each design the surface mesh representation with about  $n = 550000$  vertices has been generated. The respective performance values for the drag force  $F_D$  and rear lift force  $F_{LR}$  are linked to each design and its related mesh representation. From the entire set of designs, four different meta design datasets are compiled, containing the designs from:

- **Dataset 1:** optimized LHS with  $\mathcal{R}_A$  (process 1),
- **Dataset 2:** CMA-ES with  $\mathcal{R}_A$  (process 2),
- **Dataset 3:** CMA-ES with  $\mathcal{R}_B$  (process 3),
- **Dataset 4:** CMA-ES with  $\mathcal{R}_A$  &  $\mathcal{R}_B$  (process 2 & 3).

From the surface and performance data the disparity and performance difference matrix have been determined as depicted in Fig. 3.4. For simplicity, the wheels of the car shape have not been considered for the calculation of the shape disparities.

In each dataset, for each design  $\mathcal{M}_m$  the vertex displacements  $\delta_{i,j}^{r,m}$  have been calculated with respect to the baseline car shape  $\mathcal{M}_r$ . For simplicity, the Euclidean distance has been used to identify pairs of corresponding vertices. In addition to the displacement values, the differences in the performance numbers have been calculated, with  $\phi_{F_D} = \phi_{F_D}^{r,m} = F_D^m - F_D^r$  and  $\phi_{F_{LR}} = \phi_{F_{LR}}^{r,m} = F_{LR}^m - F_{LR}^r$ .

The shape of the reference design and its related surface mesh are illustrated in Fig. 3.5.

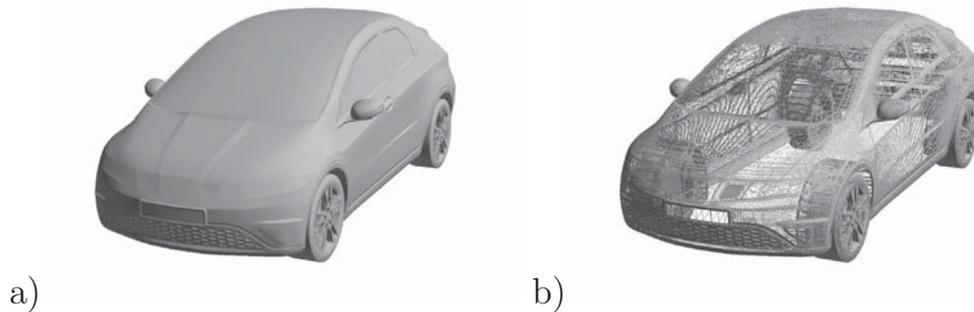


**Figure 3.4:** Illustration of the disparity and performance difference data generation process for the passenger car design data.

### 3.5.1 Approaching Design Similarities

An estimation of the the local surface differences between designs, e.g., using the displacements, allows to examine design variations that are the cause for the change in the design quality. Comparing two individual designs can already provide valuable domain information. As an example, taking the overall best design and comparing the local surface differences with the initial design provides insights about the cause for the performance raise. For illustration, as depicted in Figs. 3.6, two designs a) and b) have been selected from all design datasets. Where design a) and b) represent examples of high and low aerodynamic performance, which result from design process 3 and 1, respectively. A design is supposed to outperform other designs if the drag and rear lift force are low in comparison.

The shape of the selected designs are shown in Figs. 3.7. To illustrate the local differences of the designs, the displacements are calculated with respect to the baseline passenger car shape. The displacements are coded onto a color axis and visually projected onto the shape of the baseline designs, shown in Fig. 3.7. Blueish (redish) colored areas have been moved to the inside (outside) of the baseline shape opposite (in the direction) to the direction of the vertex normals. Thus, the graphics provide a visual impression of the design changes, revealing possible causes for the difference in the overall



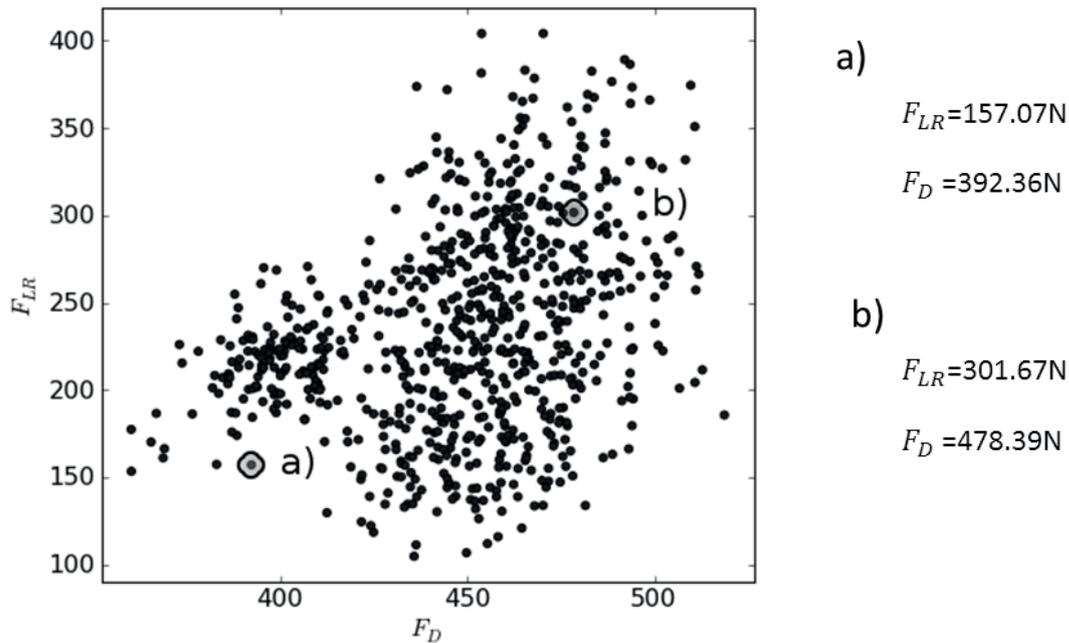
**Figure 3.5:** *Illustration of the baseline shape a) and its related surface mesh representation b).*

design quality. On the one hand it can be observed from Fig. 3.7 a) that surface regions at the top, front and side of the car have been changed to the inside resulting in a well performing car. On the other hand if surface regions at the top and side of the car are deformed to the outside the performance has dropped significantly. Engineers can utilize such information to compile new hypothesis about the design domain and validate the hypothesis in subsequent experiments.

While the analysis of individual solutions gives specific insights into the design space, an analysis and integration of the information in larger design datasets allows to identify more general coherences, e.g., for identifying weakly deformed design regions or to evaluate the course of a design process.

#### 3.5.2 Identifying Weak Deformed Design Spots

The analysis of the local disparity data, without considering the variations in the design quality, tells already a lot about the underlying design synthesis process. Basic statistics of the local surface differences provide information about the exploration of the high-dimensional design space, regarding the holistic design. Thus, evaluating relative standard deviation with respect to a chosen reference design allows the identification of areas which have rarely been modified during the design process. For an efficient design process an extensive exploration of all feasible design variations is important. Depending on the complexity, design representations used and the time horizon of the synthesis process, engineers might be more or less aware of all variations that have been applied to the car shape. The calculation of the variance over the local surface differences in a dataset, accumulated during the design synthesis, allows to study and illustrate the applied shape variations. Due

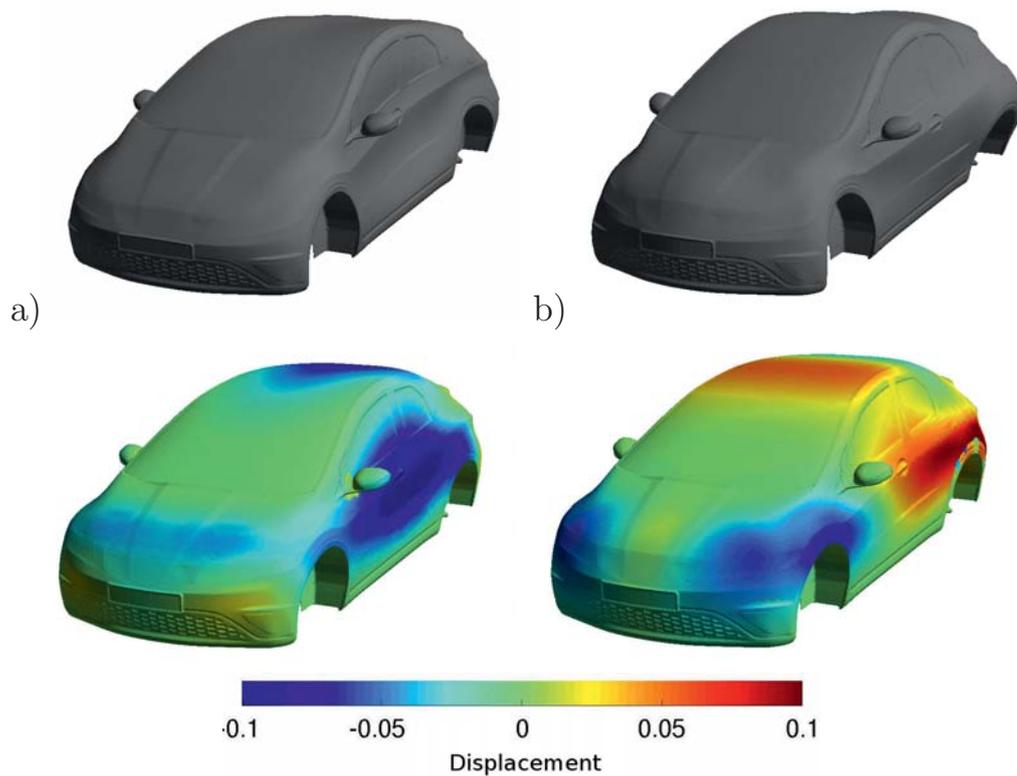


**Figure 3.6:** Selection of two designs a) and b) for exemplifying the investigation of individual design variations. Design a) and b) represent examples of high and low aerodynamic performance.

to the use of the universal design representation the analysis can be carried out for individual as well as combined process data.

The analysis of the variance of local surface variations has been applied to the displacement data resulting from the example synthesis process of the passenger car. The surface mesh representing the initial shape of the passenger car has been used as reference mesh for all individual analysis. Figs. 3.8 depict the results related to four different datasets: a) design process 1, using LHS based on representation A, b) design process 2, using CMAES and representation A, c) design process 3, using CMAES and representation B and d) a combination of design process 2 and 3, using both CMAES but different representations A and B. The individual variance values assigned to each vertex have been mapped onto color values and projected back onto the car surface. Bluish areas indicate no or weakly deformed surface areas, whereas reddish areas highlight those regions with high variance.

From Fig. 3.8 a) it can be seen that the constraints on the representation, e.g., at the front screen result in a low variance. Comparing Figs. 3.8 a) and b) one can note: while the LHS strategy targets an equal variation of all design parameters, what is also visible from the analysis of the local surface differences, the CMAES exploits a more restricted regime of surface areas, due to the adaptation of the optimization strategy. Fig. 3.8 illustrates the analysis of the combined datasets. It allows to get a wider view on the

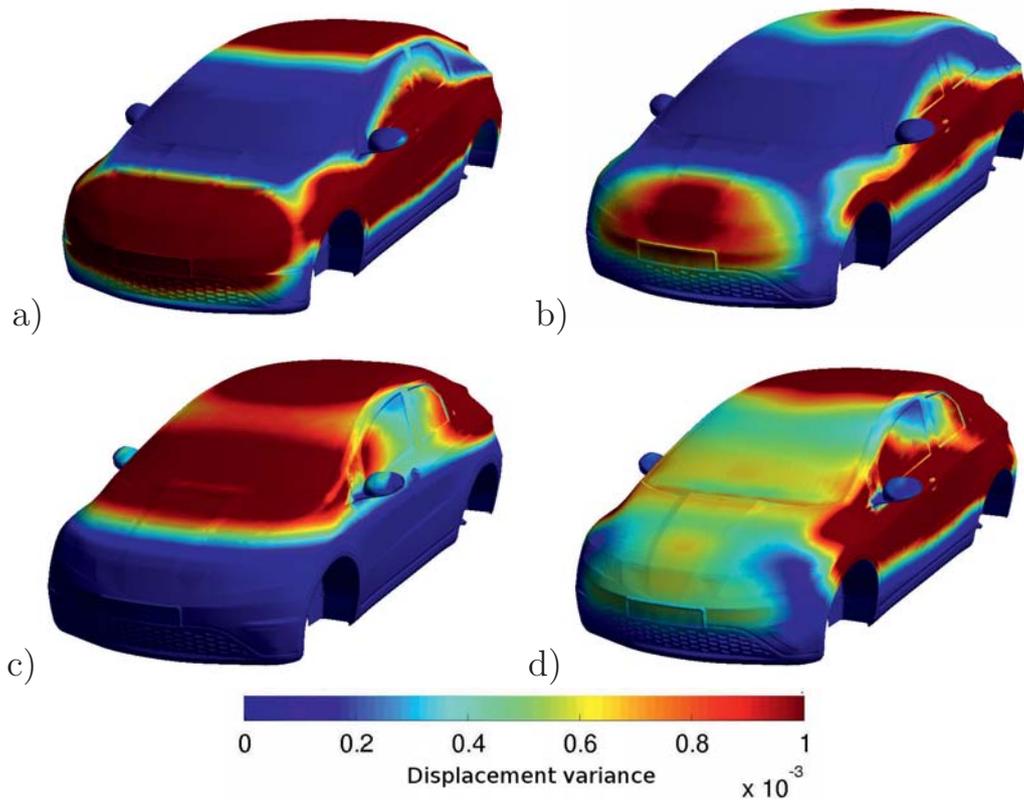


**Figure 3.7:** *Illustration of the displacements of two selected designs a) and b) relative to the baseline design, where a) is a design with high and b) with low aerodynamic performance.*

variations applied, e.g., if the design synthesis is split into multiple design processes.

A low variance might be assigned to a vertex for several reasons, either the variation of the vertex has not been feasible due to limitations in the representation, e.g., due to hard constraints on the shape variations, or those variations were not considered in the design process. On one hand, if the low variance is due to design restrictions, one might think about a change of the representation for subsequent processes, considering the exploration of alternative degrees of freedom (limited by the constraints). On the other hand, if the low variance is due to the course of the design process one might re-think about the design strategy within sub-sequent processes.

Also the variance analysis has been carried out based on offline data, the variance map could be continuously updated during the course of the synthesis process, providing information about the exploration of the feasible design variations at any point in time.

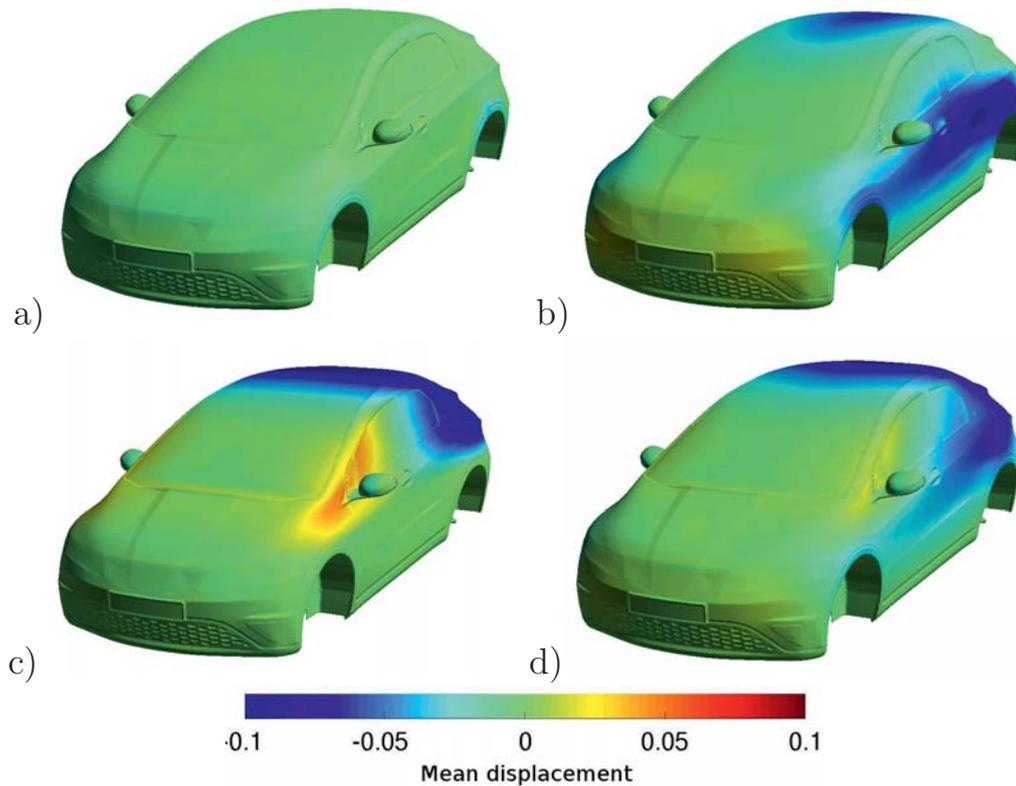


**Figure 3.8:** Visualization of the displacement variance based on data from different design processes: a) Dataset 1 to d) Dataset 4.

### 3.5.3 Evaluating the Course of Design

Given the initial objectives and constraints of the synthesis process, each design process implements a certain strategy in order to reach the design goals. A clear strategy might be obvious for individual design processes and for distinct design steps, but for multiple sequential and parallel design processes, where many engineers are involved, the actual direction of the global synthesis process might not be apparent to everyone. The integration of the local surface differences, by means of calculating, e.g., the mean vertex displacement from different design processes can reveal the actual strategy. Estimating the arithmetic mean of the displacements for each vertex relative to a pre-defined reference design provides information about the direction of the vertex displacement with respect to the vertex normal, and the amount of vertex displacement expected.

The direction of the design processes, by means of calculating the mean vertex displacements, are derived from the analysis of the four different design datasets, resulting from three distinct processes. The mean values are coded onto a color map. Reddish (bluish) regions show that the average displacement from the reference design is in (against) the direction of the



**Figure 3.9:** Visualization of the mean displacement values based on data from different design processes: a) Dataset 1 to d) Dataset 4.

surface normals, towards the outside (inside) of the car surface. Greenish areas depict regions where the average displacement is zero, indicating that these surface areas haven't been modified (see the variance analysis) or the displacements in either directions have canceled each other out. The results of the calculations are depicted in Fig. 3.9 a) to d).

Since the LHS targets a unified variation of the designs, the mean displacement value for each vertex vanishes, and an explicit direction or strategy of such sampling process is not visible, as intended by the sampling technique, see Fig. 3.9 a). Small deviations from a zero mean displacement can be observed for individual vertices, which is due to the transformation from control point parameter variations to variations of the surface points. Figs. 3.9 b) and c) depict the overall direction of the optimization algorithm that has been followed. Using the representation A (Fig. 3.9 b)), the distinct blue surface areas have been moved to the inside of the car shape in order to minimize the overall aerodynamic drag, while keeping certain restrictions onto the rear lift of the car. While the optimizer using representation A mostly tried to modify parts of the surface to the inside, the optimizer using



## CHAPTER 3. META DESIGN REPRESENTATION

---

representation B deformed the surface patches around the side mirror to the outside of the surface (averaging over all design variations), see Fig. 3.9 c).

Due to the unified representation of the design shapes, an analysis of the combined direction of design process 2 and 3 is feasible, as shown in Fig. 3.9 d). The image illustrates the mean direction of both processes.

Hence, the information about the mean design variations can be used to control and revise local and global strategies of individual and overall design processes equally. Finally, it needs to be considered, that given the same design data, the mean displacement values can change with the selection of a different reference design. Nevertheless, the different mean displacement values denote the same information just from a different "viewing angle".

### 3.6 Interim Summary

---

In this chapter the unstructured surface mesh has been introduced as a universal meta representation for object shapes. The unstructured surface mesh builds up a high dimensional geometrical approximation of the continuous object surface. The derived object representation is independent of the representation used to form the designs in different design and optimization processes.

Based on the surface mesh representations of various design shapes, the estimation of local feature disparities between mesh pairs allows to study the differences between designs in more detail, e.g., concerning vertex displacements or curvature differences. The estimation of the feature disparities requires algorithmic steps for:

- feature extraction,
- estimation of corresponding vertices and
- the calculation of the actual disparities

to be implemented. The transformation of the design shapes into surface meshes and the evaluation of the design disparities are pre-processing steps taken as a prerequisite for the shape mining process.

The investigation of the shape disparities can already reveal important information about the designs and their synthesis process. With the application of the suggested technologies and algorithms to the passenger car design data, it has been demonstrated how the analysis of the disparity data can be utilized to:



### 3.6. INTERIM SUMMARY

---

- investigate individual design solutions,
- identify weakly explored design regions and
- evaluate the course of the design process.

In this respect, the analysis of the surface mesh disparities provides a holistic view on the designs and their related synthesis processes.





# 4

## Design Sensitivity Analysis

As has been shown in Chapter 3, the transformation of the surface representation into a universal meta-design representation together with the evaluation of local shape features provides an holistic view on the design data. The investigation of the surface features and related data statistics facilitates the study of the engineers design intents and the course of optimization cycles. However, an explanation and understanding of design feature variations concerning their effect on the overall design performance has not been considered. In order to analyze those interdependencies, this chapter introduces methods for *sensitivity estimation*. State of the art methods are reviewed and a robust variant of the mutual information index is studied in more detail, which allows to reliably estimate linear and non-linear interrelations for a limit number of designs. The results from the sensitivity estimation can be utilized to filter potentially irrelevant design features from the data by means of identifying *sensitive design areas*. Exploiting sensitivity information and the geodesic distance, a new algorithm for an automatic retrieval of sensitive areas is introduced. Finally, relevant fields of application are demonstrated by applying sensitivity analysis to the passenger car design data.

### 4.1 Sensitivity Estimation

---

One of the main challenges in decision making towards an efficient design processes is the ability to predict the effect of design variations on the functional design properties. In this respect, the understanding of which design features influence the performance of a design is an essential component.

In its general definition, sensitivity analysis targets to reveal the influence of one or multiple independent variables on an dependent variable by applying methods for sensitivity estimation, see [Saltelli et al., 2007] and [Cullen and Frey, 1999]. In this context, the dependent variables are representing the cause or input of a model or system, whereas the dependent variable is



## CHAPTER 4. DESIGN SENSITIVITY ANALYSIS

---

representing its effect. Adopting the general formulation, and applying it to the analysis of design data, the following definition for the process of *design sensitivity analysis* is derived:

**Design Sensitivity Analysis** Design sensitivity analysis is the study and quantification of the impact of design feature variations on the variation of the overall design performance.

In the shape mining process, the design sensitivity analysis is carried out for two purposes. First, it is applied to extract knowledge about the importance of particular design areas with respect to variations of the design performance, and second to filter design variables in order to improve subsequent modeling and data mining steps.

A wide range of methodologies concerning the quantification of sensitivity have been studied. Those attempts can be categorized into local and global methods. *Local sensitivity estimates* aim at directly quantifying the gradient at a certain fixed point by means of calculating the partial derivative  $|\partial y / \partial x_i|_{x_0}$ , where  $y$  is the dependent variable,  $x_i$  a design variable and  $x_0$  defines the fixed point. Those methods relate to adjoint modeling [Giles and Pierce, 2000; Othmer, 2006; Othmer et al., 2011] and automatic differentiation [Verma, 2000].

Following [Saltelli et al., 2007], local methods are only informative at the position of the fixed point. In contrast to local methods, *global sensitivity analysis* methods aim at estimating the relation between input and output variables, given a limited number of data samples over a wider and closed design or input space. In [Ascough et al., 2005], the authors present the results of a qualitative evaluation of the most popular methods. Among them are the Fourier Amplitude Sensitivity Test (FAST) [Cukier et al., 1977], the Sobol' method [Sobol', 1993] and the mutual information index (MII) [Critchfield et al., 1986].

For the estimation of the total correlation, the FAST and Sobol' method apply a multi-factor analysis taking the interrelation between all independent variables into account. Although, multi-variate correlation analysis allows a more correct quantification of the sensitivities for a large number of variables, its applicability is limited due to the high computational expense. This is getting apparent when analyzing high-dimensional shape representations like unstructured surface meshes. In contrast, univariate correlation analysis provides only a qualitative estimate of the importance of design variables, but with manageable computational costs. It should be noted that using univariate correlation analysis, co-variations regarding other factors, which are not included in the analysis, act as structural noise and can bias the sensitivity estimates.

In the remainder of this chapter, correlation techniques from statistics and methods derived from the field of information theory for univariate sensitivity analysis are studied in more detail. A robust extension of the mutual information sensitivity index is studied. Opportunities and limitations of the sensitivity methods in the light of analyzing shape data with respect to aerodynamic objectives are discussed, and conclusions are drawn from the application to the passenger car design data.

### 4.1.1 Linear Correlation Techniques

Given  $\Delta_i^r$  as a measure of local design feature variation and  $\phi_i^r$  as the variation in the design performance, the design sensitivity is defined by the strength of the interrelation between  $\Delta_i^r$  and  $\phi_i^r$ . Considering a linear analysis of the relationship between design and performance variables in an analytical form, the sensitivity is defined by the differential  $\partial\phi_i^r/\partial\Delta_i^r$ .

However, the analytical function of  $\phi_i^r$  is typically not available and needs to be estimated from individual observations. There, the calculation of the slope  $\Delta\phi_i^r/\Delta\Delta_i^r$  of a fitted regression line provides a sensible approximation of the analytical sensitivity value. For statistical normalized variables the definition of the slope of the regression line is equivalent to Pearson's linear correlation coefficient [Rodgers and Nicewander, 1988], based on which the *relative vertex correlation* is defined. It quantifies the bivariate relation between the feature disparity for each vertex and the performance difference with respect to a given reference design. The relative vertex correlation is formally defined as:

$$r_{\Delta_i}^r = \frac{\sum_{m=1, m \neq r}^{N_D} (\Delta_{i,j}^{r,m} - \bar{\Delta}_i^r)(\phi^{r,m} - \bar{\phi}^r)}{(N_D - 1)\sigma_{\Delta_i^r}\sigma_{\phi^r}}, \quad (4.1)$$

where  $\bar{\phi}^r$  and  $\sigma_{\phi^r}$  are the arithmetic mean and standard deviation of the quantified performance differences, respectively. The arithmetic mean and standard deviation of the feature variables,  $\bar{\Delta}_i^r$  and  $\sigma_{\Delta_i^r}$ , have already been defined in equations (3.22) and (3.24) (Section 3.4). As for the calculation of the standard deviation, it holds that for an integration of the correlation coefficient over multiple reference designs  $\bar{\Delta}_i^r$  and  $\bar{\phi}^r$  might vanish, what can simplify the calculation of the correlation coefficient.

Based on equation (3.23) and (3.25), an update rule for the relative vertex correlation can be constructed. For  $0 < r_i^r \leq 1$  and  $0 > r_i^r \geq -1$  the increase of the feature value is expected to result in an increase or decrease of the performance value, respectively. For  $r_i^r = 0$  a change of the feature value is expected to have no effect on the objective function.



## CHAPTER 4. DESIGN SENSITIVITY ANALYSIS

---

The statistical significance of the correlation coefficient can be estimated using the students t-statistics,

$$t_{\Delta_i}^r = \frac{r_{\Delta_i}^r}{\sqrt{(1 - (r_{\Delta_i}^r)^2)/(N_D - 1)}}. \quad (4.2)$$

In the presence of outliers in the data the Pearson's correlation coefficient might not reflect the correct sensitivity of a feature variable. In such circumstances, an a priori identification and exclusion of outliers is advisable. As an alternative, *Spearman's rank correlation coefficient*  $\tilde{r}_{\Delta_i}^r$  is applicable, where the correlation coefficient is calculated based on the ranks of the feature values instead of the feature values itself. Spearman's correlation coefficient is less sensitive to outliers and results in high correlation values also for monotonic relations between feature and performance variable. Hence, relaxing the strict linearity assumption of the Pearson correlation coefficient.

The standard Pearson and Spearman correlation coefficient ignore the effect that the correlation of one feature variable can strongly be biased by others, resulting in an over or under estimation of the correlation. To overcome this shortcoming, the *partial correlation coefficient* can be applied for a multi-variate correlation analysis. It estimates the bivariate association between two variables under the control of other input variables of the system. In the context of a high dimensional surface mesh representation all but feature  $i$  are to be considered for the calculation of the partial correlation coefficient. The complexity of the partial correlation coefficient calculation is  $O(n^3)$ , with  $n$  defining the number of features linked to the respective vertices. For typical surface mesh sizes, with  $n > 1000$ , the calculation of the partial correlation coefficient becomes impractical.

### 4.1.2 Mutual Information Based Sensitivity Index

The Pearson and Spearman correlation coefficients make strong assumptions on the linearity of the interrelation either between the feature values or their ranks. The Sobol' technique, which is based on ANOVA (analysis of variance), requires to fit an underlying regression model or response surface to the given data. Thus, the results of the Sobol' algorithm strongly rely on the goodness of fit of the regression model. Information theoretic attempts like the *mutual information* quantify the association between variables by comparing the underlying data distributions directly and make no assumptions on the functional relationship between the independent and dependent variables.



## 4.1. SENSITIVITY ESTIMATION

---

The quantification of the mutual information has its origin in the article of Shannon published in 1948 [Shannon, 1948]. The basis of Shannon's information theory is the formulation of the entropy. For a discrete random variable  $X$ , Shannon's entropy is defined as:

$$H(X) = - \sum_{x \in \mathbb{X}} p(x) \log_2 p(x). \quad (4.3)$$

The measurement unit of the entropy  $H(X)$  is typically defined in bits, as long as the Logarithm is taken to the basis of two. The intention of Shannon has been to quantify the uncertainty of information, which is contained in a message transmitted from a sender to a receiver. Beyond Shannon's intention, the entropy  $H(X)$  provides a quantity suited to evaluate the expected deviation of a series of observations from a singular event guaranteed to happen. The Shannon entropy is always greater or equal to zero and reaches its maximum for a uniform random variable  $U$ , with  $0 \leq H(X) \leq H(U)$ .

It is important to note that the value of the entropy is sensitive to the number of discrete states of the random variable. For equiprobable events,  $H(X)$  increases with an increasing number of states. Given the joint probability distribution of two discrete random variables  $X$  and  $Y$ , the entropy formulation can be extended to multiple attributes as follows:

$$H(X, Y) = - \sum_{x \in \mathbb{X}, y \in \mathbb{Y}} p(x, y) \log_2 p(x, y), \quad (4.4)$$

with  $p(x, y)$  defining the joint distribution of  $X$  and  $Y$ . The difference between the joint entropy and the marginal entropy of either of them,  $H(Y|X) = H(X, Y) - H(X)$ , is known as the conditional entropy.  $H(Y|X)$  quantifies the remaining uncertainty of a variable  $Y$  if a second variable  $X$  is already known. In case of  $X$  and  $Y$  being statistically independent, the remaining uncertainty of a variable  $Y$  given  $X$  equals the uncertainty of  $Y$ ,  $H(Y|X) = H(Y)$ . Thus, the joint entropy is a linear superposition of the marginal entropies with  $H(X, Y) = H(X) + H(Y)$ .

The relationship between the marginal and joint entropies in case of statistical independence provides the basis for the calculation of the mutual information. The mutual information  $I(X; Y)$  quantifies the information shared between two variables  $X$  and  $Y$ . Under the assumption of independence, a model of the joint entropy, with  $\hat{H}(X, Y) = H(X) + H(Y)$ , is compared to the entropy calculated based on the observed joint probability masses,  $H(X, Y)$ . This led to the mathematical formulation of the mutual information:

## CHAPTER 4. DESIGN SENSITIVITY ANALYSIS

---

$$I(X; Y) = H(X) + H(Y) - H(X, Y). \quad (4.5)$$

In terms of marginal and joint probability masses the mutual information is defined as follows:

$$I(X; Y) = \sum_{x \in \mathbb{X}, y \in \mathbb{Y}} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}. \quad (4.6)$$

In case that  $X$  and  $Y$  are independent, the mutual information vanishes so that  $I(X; Y) = 0$ . In all other cases  $I(X; Y)$  takes positive values bounded by the maximum of the marginal entropies  $H(X)$  and  $H(Y)$ . With  $I(X; Y) = H(X) = H(Y)$ , the mutual information reaches its maximum value in the case, where the distributions of  $X$  and  $Y$  are identical. When applying mutual information for the purpose of sensitivity analysis [Critchfield et al., 1986], typically the mutual information is normalized by the entropy of the target variable  $Y$ :

$$s(X, Y) = \frac{I(X; Y)}{H(Y)}, \quad (4.7)$$

where  $s(X, Y)$  is the sensitivity of  $Y$  with respect to variations in  $X$ .

The information based *total sensitivity index*, see [Lüdtke et al., 2008], is a multi-variate extension of the mutual information to multiple attributes. Given a set of  $n$  attributes  $\mathcal{X} = \{X_i, \dots, X_n\}$ , and the assumption that  $Y$  is totally explained by the variables in  $\mathcal{X}$ , the total sensitivity index in terms of Shannon information is defined as follows:

$$s_{total_i}(\mathcal{X}, Y) = \frac{I(\mathcal{X}; Y) - I(\mathcal{X} \setminus X_i; Y)}{H(Y)}. \quad (4.8)$$

It quantifies the difference between the information that is transferred between all input variables and the target variable  $I(\mathcal{X}; Y)$ , and the information that is transferred between all but the considered variable  $X_i$  and the target variable,  $I(\mathcal{X} \setminus X_i; Y)$ . The total sensitivity index captures the main effect of  $X_i$  and all higher order interactions between  $X_i$  and the remaining attributes in  $\mathcal{X}$ . The calculation of  $s_{total_i}(\mathcal{X}, Y)$  requires the estimation of the  $n + 1$  dimensional joint distribution of the variables in  $\mathcal{X}$  plus  $Y$ . The estimation of the joint distribution for a high number of variables suffers from the curse of dimensionality, e.g., see [Scott, 1992, Chapt. 7]. Thus, the application of  $s_{total_i}(\mathcal{X}, Y)$  to high dimensional shape representations is infeasible, or at least requires an a priori feature reduction step.

Considering the attributes which describe the shape feature variations  $\Delta^r$  and the performance attributes  $\phi^r$  as random variables allows the information

theoretic estimation of vertex sensitivities. It requires the estimation of the marginal and joint densities based on the observed feature and performance values. An introduction into techniques for the estimation of probability densities is given, e.g., by [Scott, 1992].

### 4.1.3 Robust Estimation of Mutual Information

A reliable estimation of the probability densities for the calculation of the information based sensitivity indices typically requires a large amount of data. The demand on a robust sensitivity estimate is to provide a reliable quantity even for small as well as noisy data. In real-world applications the generation of design data is always of considerable computational expense, especially when high fidelity simulations have to be carried out to calculate the design performance. In practice, continuous variables are often discretized by calculating the relative frequencies of the parameter values with respect to fixed intervals of equal range. Due to noise in the measurement of the feature or performance values, individual parameter values might wrongly be assigned to neighboring intervals. This has a strong effect on the estimated probability distribution, especially for small sample sizes.

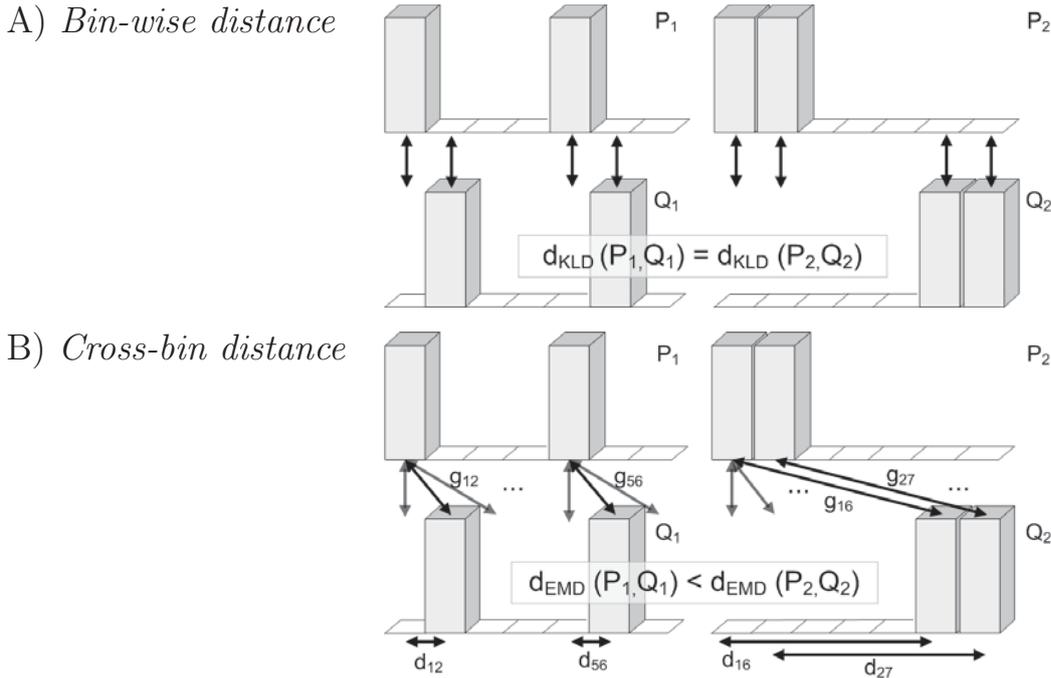
In order to reduce the effect of the discretization error on the sensitivity estimation, two strategies can be undertaken. The first approach is to adapt the method for the estimation of the probability densities, e.g., using spline functions [Daub et al., 2004], or in a second approach, directly adapt the calculation of the sensitivity index. While the first approach is typically sensitive to the specification of certain hyper-parameters, e.g., defining the degree and order of a spline function, here the second approach is investigated in more detail, targeting to come up with a parameter-free robust sensitivity estimate.

In a first step it can be observed that for two random variables  $X$  and  $Y$  the formulation of the mutual information is equal to the Kullback-Leibler divergence (KLD) [Kullback and Leibler, 1951] between the joint probability distribution  $p(x, y)$  and its Kirkwood superposition approximation (KSA) [Jakulin and Bratko, 2004a],  $\hat{p}(x, y) = p(x)p(y)$ :

$$I(X; Y) = D_{KL}(p(x, y) || \hat{p}(x, y)). \quad (4.9)$$

The KSA is an approximation of the observed joint probability distribution under the assumption of independence between the random variables. From this perspective, the mutual information can be understood as a measure of the difference between two probability distributions, where the KLD is defined as the arithmetic mean of the bin-wise distances.

## CHAPTER 4. DESIGN SENSITIVITY ANALYSIS



**Figure 4.1:** Visualization of the difference between the bin-wise a) and cross-bin b) distance. The EMD cross-bin distance is defined by a weighted superposition of the grounding distances  $d_{ij}$  and the mass differences  $g_{ij}$ .

The calculation of the bin-wise distance does not take any quantification errors into account. Therefore, in a second step the KLD is replaced by a cross-bin distance measure, e.g., by the Earth Mover's Distance (EMD) [Rubner et al., 2000]. In contrast to the bin-wise distance, the cross-bin distance takes the samples from neighboring intervals into account. Thus, for continuous variables, the cross-bin distance measure allows a more robust estimation of the distance between two distributions, especially for a low number of observations. This is depicted in Fig. 4.1.

The standard KLD, Fig. 4.1 a), would estimate the same distance between distributions  $(P_1, Q_1)$  and  $(P_2, Q_2)$ . However, the difference between  $(P_1, Q_1)$  could result from measurement noise or from a limited number of data samples, while the difference between  $(P_2, Q_2)$  seems to be more distinct. Therefore, an intuitive measure should rate the distributions  $(P_1, Q_1)$  being more similar than  $(P_2, Q_2)$ . The cross-bin distance measure, Fig. 4.1 b), does this by taking neighboring bins into account. As a result small perturbations in the data do not have a drastic effect on the cross-bin distance measure (EMD).

In [Rubner et al., 2000], Rubner first introduced the EMD as a cross-bin distance measure for estimating the similarity of color and texture histograms in two dimensional images. It has been argued that the EMD better mimics

the human perception of texture similarities. In this context, the EMD has successfully been applied to tasks like image retrieval, image registration or feature extraction, see [Sandler and Lindenbaum, 2011].

The EMD, as it has been defined by Rubner in [Rubner et al., 2000], has no closed mathematical solution, rather the EMD value is the solution to an optimization task. It is defined as the minimal total ground distance weighted by the amount of probability masses that have to be moved, so that the two compared histograms coincide. For the calculation of the EMD a linear optimization problem has to be solved. Given two discrete distributions  $P$  and  $Q$ , as well as a distance matrix  $D$ , the objective for the solution of the EMD for discrete probabilities is defined as,

$$E(P, Q, D) := \min \sum_{ij} d_{ij} g_{ij}, \quad (4.10)$$

where  $g_{ij}$  defines the flow of probability masses and  $d_{ij}$  the ground distance between bin  $i$  of distribution  $P$  and bin  $j$  of distribution  $Q$ , see Fig. 4.1 b). The optimization is subject to,

$$g_{ij} \geq 0 \quad (4.11)$$

$$\sum_j g_{ij} \leq p_i \quad (4.12)$$

$$\sum_i g_{ij} \leq q_j \quad (4.13)$$

$$\sum_{ij} g_{ij} = 1, \quad (4.14)$$

where the constraint in equation (4.11) ensures that probability masses are only moved from distribution  $P$  to distribution  $Q$ . The additional constraints, defined in equations (4.12), (4.13) and (4.14), guaranty that only as much as probability masses are moved as exist, and that all sources are depleted. After an optimal solution has been found, the distance is calculated as follows:

$$D_{EMD}(P, Q) = \frac{\sum_{ij} d_{ij} g_{ij}}{\sum_{ij} g_{ij}}. \quad (4.15)$$

The EMD makes sure that perturbations of the attribute values are not penalized excessively.

Applying the EMD instead of the KLD to equation (4.9) leads to the robust variant of the mutual information for the estimation of sensitivities:

$$RI_{EMD}(X;Y) = D_{EMD}(p(x, y), \hat{p}(x, y)). \quad (4.16)$$

The robust measure of sensitivity  $RI(X, Y)$  can not be considered anymore as an information quantity in the sense of Shannon, rather it is implementing an alternative measure of correlation. The EMD is a real distance measure and thus  $RI(X; Y) \geq 0$ . Since the solution to the EMD is an optimization problem, the calculation of the EMD is computational expensive, with a complexity of  $O(B^3 \log B)$ . The computational time increases polynomial with the number of discretization intervals. The high computational costs often hinder the implementation of the EMD to real world applications.

In [Shirdhonkar and Jacobs, 2008], the authors proposed an algorithm for the fast calculation of the EMD. The authors found a closed solution of the EMD in the wavelet domain and termed this technique wavelet EMD or shortened WEMD. For the calculation of the WEMD the difference histogram between two distributions is calculated,  $\Delta P(S) : \delta p_k = \hat{p}_k - p_k$ . After that the difference histogram is transferred into the wavelet space. In the wavelet space the actual optimization task is reduced to a weighted sum of wavelet coefficients,

$$D_{WEMD}(P, Q) = \sum_{\lambda} 2^{-j(1+\frac{n}{2})} |\delta p_{\lambda}|, \quad (4.17)$$

where  $n$  is the dimension of the discrete probability distribution and  $\delta p_{\lambda}$  are the wavelet coefficients of the transformed difference histogram  $\Delta P$ . The index  $\lambda$  is used for both shifts as well as the scale  $j$  wavelet coefficient. For the wavelet transformation an wavelet function which has at least one derivative has to be chosen. Thus, e.g., the famous Haar wavelet can not be used for the calculation of the EMD. The complexity of the WEMD increases only linearly with the number of bins  $B$ . Replacing the EMD with the WEMD provides means for a more efficient estimation of the cross-bin distance based sensitivity:

$$RI_{WEMD}(X;Y) = D_{WEMD}(p(x, y), \hat{p}(x, y)). \quad (4.18)$$

## 4.2 Identification of Sensitive Areas

---

Global sensitivity analysis based on surface data provides the engineer with information about the relevance of local surface deformations to a given design performance. There, the number and locality of the considered surface variations depends strongly on the surface sampling density. Typically, the

## 4.2. IDENTIFICATION OF SENSITIVE AREAS

---

number of vertices that build the discrete surface mesh is huge ( $n \gg 1000$ ) to ensure an accurate approximation of the continuous design surface.

In practice, in order to model higher order relationships between distant design variations and the performance, a low dimensional manifold in the input space is favored. The number of input parameters needs to be small to produce reliable computational models mapping design variations onto the performance values. While individual modeling techniques comprise automatic parameter selection or weighting, here an explicit parameter reduction step aligned to the given shape representation is introduced.

The reduction of the input dimension requires to take valid assumptions on the given data. Concerning the design data, treading a typical design process, a modification of single vertices is supposed to be unlikely, given a uniform and sufficiently accurate sampling of the surface. Rather entire design regions of a certain extent, enclosing multiple vertices, are considered for shaping new designs. Thus, taking proximity and the sensitivity of vertices into account, neighboring vertices can be grouped into *sensitive design areas*, which are formally defined as follows:

**Sensitive Design Area** A sensitive design area  $\mathcal{A}$  is defined as a subset of vertices  $\mathcal{A} = \{\vec{v}_1, \dots, \vec{v}_l\}$  of the entire mesh  $\mathcal{M}$ , which form a closed area on the surface and are similar in their sensitivity to the considered design quality.

Given a pre-defined similarity or distance measure  $d_{i,j} \mapsto f(d_{i,j}(\mathcal{M}), d_{i,j}(\mathcal{S}))$ , vertices  $\vec{v}_i$  and  $\vec{v}_j$  belong to one and the same design area if they are close with respect to spatial distance  $d_{i,j}(\mathcal{M})$  and if they share a similar sensitivity value  $d_{i,j}(\mathcal{S})$ .

For different design performance values, different sensitive areas might exist. Each sensitive design area can be treaded as a single parameterized unit for a subsequent modeling step. The observed change of the properties at the position of the vertex that is located in the center of each sensitive area  $\mathcal{A}$  denotes changes of the entire area. Hence, the number of sensitive design areas defines the dimensionality of the input space for the subsequent construction of computational models.

### 4.2.1 Unsupervised Clustering

Clustering techniques are applicable to automatically derive sensitive areas in a unsupervised manner. Following [Jain et al., 1999], "*Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters).*". In the related publication a comprehensive



## CHAPTER 4. DESIGN SENSITIVITY ANALYSIS

---

overview on the clustering concept, a taxonomy of clustering techniques and relevant applications related to image segmentation, object recognition and information retrieval are described. As already summarized in [Jain and Dubes, 1988], the application of clustering algorithms usually involves the consideration of:

- the representation of features or patterns,
- the definition of the proximity,
- the actual grouping of patterns and
- eventually the abstraction of data and the assessment of the clustering results.

Furthermore, the normalization of data and the efficiency of the algorithm chosen have to be considered. Beside the choice of an appropriate clustering algorithm, the definition of the similarity measure is crucial.

The most frequently used similarity measure is the Euclidean distance  $d_2$ , as a special case of the Minkowski distance  $d_p$ . The Euclidean distance is shown to be applicable if the data set constitutes compact or isolated clusters, but is strongly dominated by the largest-scaled feature [Jain et al., 1999; Mao and Jain, 1996]. In [Rokach and Oded, 2005], the authors provide an overview of alternative distance measures for the calculation of pattern similarity. However, often the applied similarity measure has to be tuned to the given data characteristics. In some cases adding domain information to the features, to the distances measure or to the clustering strategy can be beneficial.

The diverse set of clustering methods that have been developed over time can roughly be grouped into hierarchical and partitioning methods. *Hierarchical clustering algorithm*, which itself can be separated into single-link, complete-link and average-link algorithms, recursively partition the data set into nested groups of patterns. In contrast, *Partitioning clustering algorithms* obtain a single partitioning of the data samples into different clusters. A review on the different algorithms can be found in [Jain et al., 1999; Rokach and Oded, 2005].

One of the most frequently used techniques is the k-means clustering algorithm [Duda and Hart, 1973]. The algorithm starts with randomly defining  $k$  cluster centroids. Iteratively, all patterns are assigned to its closest centroid followed by a recalculation of the new centroids and so forth, until a convergence criteria has been met. Typically, the iterative procedure is stopped when the location of the centroids remains constant. The K-means approach

is supposed to identify a local optimal cluster configuration in the data. K-means exploits the pre-defined distance measure to assign patterns to their closest centroid.

The reason why the k-means algorithm has such a high popularity is due to its simplicity and its low computational complexity. The algorithm scales linearly with the number of clusters, number of data samples and its dimensionality [Selim and Ismail, 1984]. Beside the advantages, some major drawbacks should be mentioned as well. First, the number of clusters has to be pre-defined. An automatic identification of the number of clusters is not realized. Second, the results of the K-Means clustering procedure strongly depend on the choice of the initial centroids. Therefore, often different clustering runs with different initializations have to be carried out in order to achieve a proper result. The K-Means algorithm typically works well if isotropic clusters can be fetched from the data [Rokach and Oded, 2005].

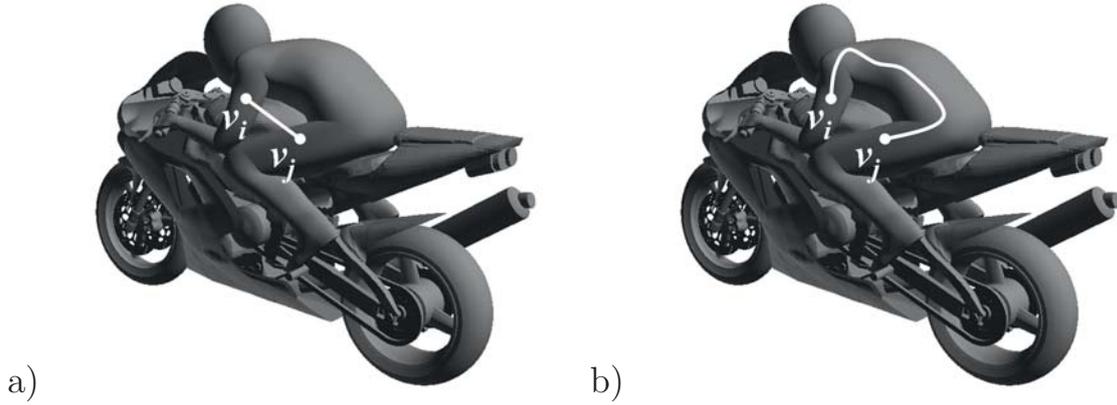
Using more sophisticated techniques, like the X-Means clustering [Pelleg and Moore, 2000] or the gap-statistic [Tibshirani et al., 2001], can overcome the problem that the number of clusters needs to be defined in advance. The gap statistic is applicable to the results of any clustering algorithm and any distance measure. It defines a score for choosing the correct number of clusters from multiple experiments carried out with a different cluster number.

### 4.2.2 Measure of Vertex Similarity

Related to the definition of a sensitive area, the similarity between two vertices  $\vec{v}_i$  and  $\vec{v}_j$  on mesh  $\mathcal{M}$  is defined based on the spatial distance between two vertices  $d_{v_i, v_j}(\mathcal{M})$  and the similarity in the sensitivity  $d_{i, j}(\mathcal{S})$ . The spatial distance  $d_{v_i, v_j}(\mathcal{M})$  measures the distance between vertices in the Euclidean space based on the coordinates of the surface points. Two distance measures are considered and discussed here.

One attempt calculates the Euclidean distance between vertices, measuring the distance in the three dimensional space in which the surface mesh is embedded. However, the Euclidean distance might underestimate the distance between vertices on the surface especially around areas of high curvature. Vertices which are close in the three dimensional space do not have to be close along the two dimensional manifold of the surface.

Therefore, in a second attempt the *geodesic distance* which estimated the distance between vertices along the shape surface is used. It is expected that the geodesic distance better reflects the correct spatial distance between vertices. Hereafter, the metric exploits the connectivity of the surface mesh representation.



**Figure 4.2:** Indication of the difference between calculation of the Euclidean distance a) and the Geodesic distance b), applied to the motorbike mesh (Example mesh supplied with OpenFOAM, the open source CFD simulation software).

In a pre-processing step, the surface mesh is normalized into the unit cube. All pairwise geodesic distances  $d_{i,j}^g$  are calculated by identifying the shortest path between two points in the mesh using the algorithm of Dijkstra, see [Dijkstra, 1959]. The distances of the edges are summed up along the shortest path to estimate the geodesic distance. Given the surface mesh of a motor bike, Figs. 4.2 show the conceptual difference between the Euclidean and the geodesic distance.

In addition to the spatial distance the difference between the normalized sensitivity values  $d_{i,j}^s$  is calculated. A crucial task is the superposition of the spatial distance and the sensitivity similarity. In order to avoid the selection of a proper weighting, an overall similarity measure is chosen that implements the following heuristics. If the same sensitivity value is assigned to both vertices  $\vec{v}_i$  and  $\vec{v}_j$ , the overall similarity should equal the spatial distance. If a maximum difference between the sensitivities is measured, the similarity measure is expected to increase the overall similarity based on the spatial distance by a factor of two. Thus, given that the sensitivity values are normalized between zero and one, the overall similarity measure is defined as follows:

$$d_{i,j} = d_{i,j}(\mathcal{M}) \cdot (1 + d_{i,j}(\mathcal{S})), \quad (4.19)$$

where  $d_{i,j}(\mathcal{M})$  is considered to be either the Euclidean or geodesic distance.



### 4.2.3 Automatic Identification of Sensitive Areas

The combination of the unsupervised clustering techniques with the introduced measure of vertex similarity facilitates an automatic retrieval of sensitive areas. Adapting the typical clustering procedure, e.g., see [Jain and Dubes, 1988], the following steps constitute the automatic procedure for identifying sensitive areas:

1. Definition of the similarity measure (including spatial and sensitivity information)
2. Pre-select vertices based on their sensitivity (optional)
3. Select and apply the unsupervised clustering procedure to derive sensitive areas
4. Determine the cluster centers
5. Assign the local surface differences of the cluster centers and the design related properties to a reduced dataset

Before the actual clustering, non-sensitive vertices can be filtered from the entire set of vertices. This simplifies the clustering and reduces computational costs. The definition of which vertices are non-sensitive depends on the sensitivity measure chosen. Statistical significance tests provide a general attempt to reject non-sensitive vertices, quantifying the evidence that the vertex is not defined as sensitive by chance. If the significance level of the calculated sensitivity is below a certain p-value, e.g.,  $p = 0.1$  (10%), the vertex is rejected. Based on the t-statistics, see equation (4.2), the p-value can be derived for the Pearson and Spearman correlation coefficient. For the calculation of the statistical significance of the mutual information based sensitivity, e.g., see [Dawy et al., 2006].

A possible alternative to the statistical significance test is the definition of a fixed sensitivity threshold  $\lambda$ , so that all vertices which have a sensitivity value below  $\lambda$  are considered to be insensitive. However, the choice of an appropriate threshold is a difficult task. For all sensitivity methods a stringent division between sensitive and insensitive values does not exist.

In [Cohen, 1988], Cohen tried to derive a classification of the linear correlation coefficient based on psychological experiments. With the results of this study one could state that a minor or no linear correlation is observed if the absolute value is below 0.3. Thus, concerning the linear correlation coefficient, a threshold set to  $\lambda = 0.3$  might be a sensible choice. However, the discrete classification of the correlation value is still vague.

Without loss of generality, k-means clustering is applied to derive  $k$  groups of vertices on the design surface. The vertices closest to the arithmetic cluster centers and their related features provide an abstract representation of each sensitive area.

### 4.2.4 Example

Fig. 4.3 illustrates an artificial example, outlining the basic idea for the identification of sensitive areas. In the example, the reference design is defined by the triangulated surface mesh of a simple cube. An arbitrary set of vertices has been selected from the mesh for clustering. The goal of this example is to demonstrate the difference of the automatic surface clustering with respect to different distance measures.

In Figs. 4.3 a), b) and d), vertices which belong to the dark blue area, labeled with  $\mathcal{A}_0$  ( $A0$ ), are not considered for clustering. Figs. 4.3 a) and b) show the clustering results using solely the Euclidean a) and the geodesic distance b), in order to measure the vertex similarity. K-means clustering, with  $k = 3$ , has been applied for grouping the vertices on the surface using either of the two spatial distances. It can be seen that using the geodesic distance results in distinct closed areas  $\mathcal{A}_1$  to  $\mathcal{A}_3$ , while using the Euclidean distance results in a separation of area  $\mathcal{A}_2$ .

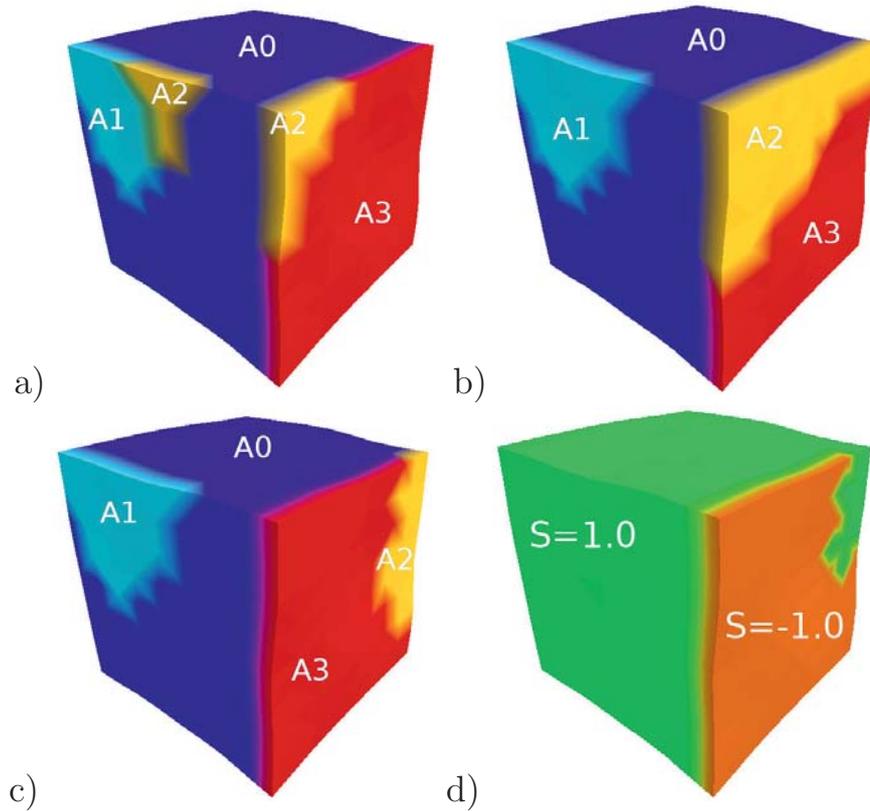
As depicted in Fig. 4.3 c) artificial sensitivity values of  $S = 1.0$  (green) and  $S = -1.0$  (orange) are assigned to the vertices. Using the distance measure from equation (4.19), which is defined by the superposition of the sensitivity information and the geodesic distance, results in a different clustering, shown in Fig. 4.3 d). The clustering algorithm groups vertices with the same sensitivity value into the same group of vertices. This can be observed by comparing areas  $\mathcal{A}_2$  and  $\mathcal{A}_3$  in Figs. 4.3 b) and d).

## 4.3 Utilizing Sensitivity Information

---

The sensitivity analysis can provide engineers with new insights into the cause of changes in the design performance. It allows engineers to predict the function of a design based on planned shape variations. With that allowing engineers to estimate the expected design performance before making any design decision in the design process.

In [Graening et al., 2008], the validity of the calculated sensitivities has been tested given the data of turbine blade geometries. Direct manipulation of free form deformation (DMFFD) [Hsu et al., 1992] has been applied to



**Figure 4.3:** Results of a simple example, where surface vertices are clustered based on: a) the Euclidean distance, b) the geodesic distance and c) the combination of the geodesic distance with sensitivity information. The corresponding sensitivity map is shown in d), where green denotes high and orange low vertex sensitivity.

create concrete shape deformations for the comparison of the predicted and the evaluated aerodynamic properties of the designs.

Furthermore, sensitivity information can be utilized to construct an initial object representation for subsequent computational optimizations. In [Graening et al., 2012], based on free-form deformation techniques, the authors have shown that an adaptation of the initial representation based on sensitivity information can lead to an improved optimization process.

## 4.4 Application to Passenger Car Design

In the following experiments sensitivity analysis has been applied to the passenger car design data. Different methods for sensitivity estimation have been applied in order to explicitly evaluate the relevance of local shape modifications captured by  $\Delta_{i,j}^{r,m}$  to the performance difference values  $\phi_{FD}$  and  $\phi_{FLR}$ , previously defined in Section 3.3. Beside the quantification of the rele-

vance, the analysis of the sensitivities, e.g., can provide qualitative information about the direction of shape modifications which are expected to result in an improvement of the design performance. Exploiting clustering techniques together with the measure of vertex similarity, sensitive areas along the passenger car surface have been identified. From the sensitive areas a reduced representation of the design surfaces has been derived.

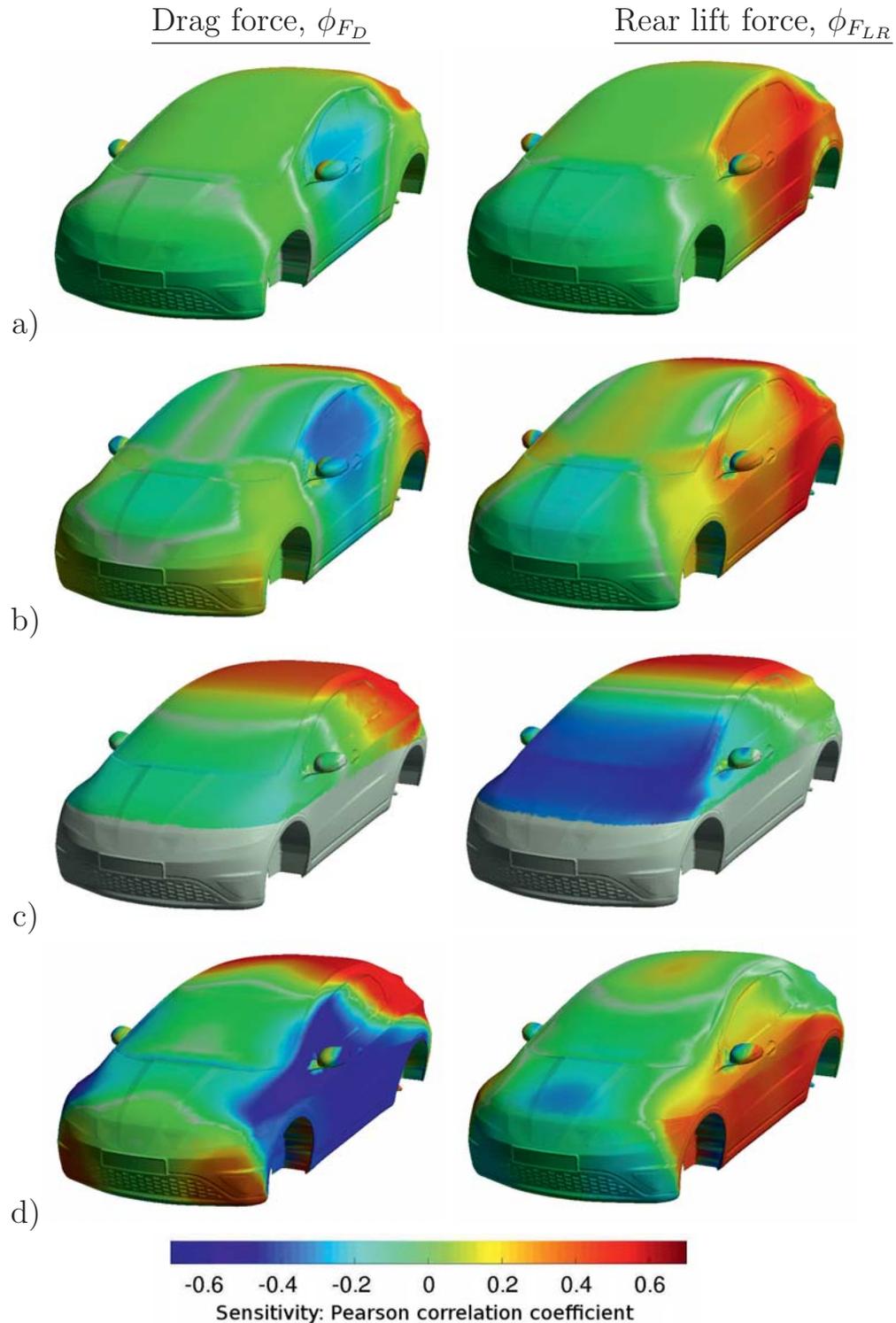
### 4.4.1 Direction of performance improvement

Given the displacement data for each vertex of the reference mesh and the differences in the performance numbers, the sensitivities for a chosen reference design have been calculated using the Pearson correlation coefficient. In addition to the correlation value the statistical significance for each correlation value has been calculated. For visualization, both values have been mapped onto one color scale, where the actual color value is defined by the correlation value and the saturation of the color is defined by its significance. The interpretation of the correlation values needs to be done closely linked to the meaning of the surface features. With respect to the vertex displacements the resulting color values are interpreted as follows:

- **Red:** The deformation of the surface patch in the direction of the normal vector is expected to *increase* the performance difference  $\phi_{F_D}$  or  $\phi_{F_{LR}}$ , or vice versa;
- **Blue:** The deformation of the surface patch in the direction of the normal vector is expected to *decrease* the performance difference  $\phi_{F_D}$  or  $\phi_{F_{LR}}$ , or vice versa;
- **Green:** The deformation of the surface patch in or against the direction of the normal vector is expected to have *no effect* on the performance difference  $\phi_{F_D}$  or  $\phi_{F_{LR}}$ ;
- **White:** Since the correlation value is insignificant, a statement about the effect of the deformation of the patch on the performance difference indicators cannot be made.

The results of the sensitivity analysis for the different design datasets 1 to 4 are depicted in Figs. 4.4 a) to d). The different figures show the same frontal view on the reference design. The sensitivities of the rear view are depicted in the appendix in Section A.1. The Figs. 4.4 on the left and right side allow a comparison of the sensitivities to  $\phi_{F_D}$  with the sensitivities to  $\phi_{F_{LR}}$  based

#### 4.4. APPLICATION TO PASSENGER CAR DESIGN



**Figure 4.4:** Results of the sensitivity analysis based on the calculation of the Pearson correlation coefficient calculated for different passenger car design datasets 1 to 4, shown in row a) to d). On the left side the sensitivities with respect to the drag force  $\phi_{F_D}$  and on the right side the sensitivities with respect to rear lift force  $\phi_{F_{LR}}$  are depicted. The respective rear view on the shape of the passenger car is depicted in appendix A.1.



## CHAPTER 4. DESIGN SENSITIVITY ANALYSIS

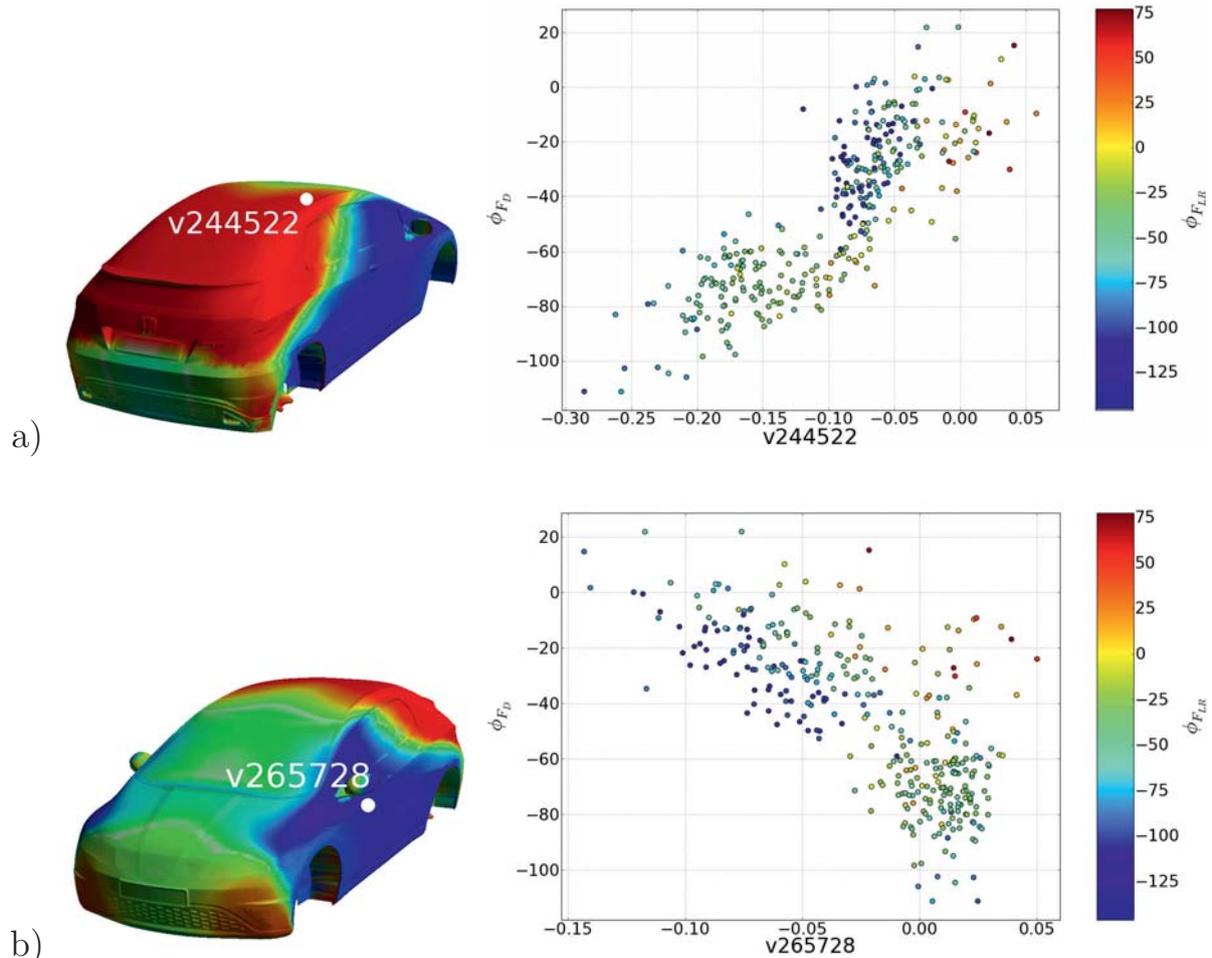
---

on the same design datasets. For dataset 1, 2 and 4, a significant trade-off between the sensitivities to the drag and rear lift force can be observed around the front door of the car shape. The deformation in the normal direction to the outside of the car shape is expected to result in a decrease of the drag force. However, such a modification would always go along with an increase in the rear lift. This indicates that with the modification of this area a minimization of both performance numbers cannot be achieved, and a trade-off solution between drag decrease and rear lift increase has to be found.

From the analysis of the results that show the drag force sensitivities, on the left hand side of Fig. 4.4, one can derive the basic rule that a deformation of the rear part to the inside of the car will reduce the drag and thus improve the car performance. While this concept is likely to be known to the aerodynamic engineer, the information that the deformation of the area close to the front door of the car towards the outside of the initial car surface can lead to an reduction of the drag denotes a more interesting relation. Furthermore, the analysis of the joint dataset (see Fig. 4.4 d)) yields to drag sensitivities at the outer front bumper, which are not obvious from the analysis of the individual process datasets (cp. Figs. 4.4 b) and c)).

Concerning dataset 3, it should be recalled that only parts of the design are modified during the optimization. From the results of Fig. 4.4 c) it can be seen that surface areas which haven't been modified will result in insignificant correlation values. Those regions have the lowest color saturation and therefore are highlighted in white. Without coding the significance values onto the color scale, such information would remain hidden.

In general, the sensitivity information from the calculation of the linear correlation coefficient not only provides engineers with information about the most relevant vertices, but together with the displacement features provides knowledge about the direction of vertex modifications that are expected to improve a design. However, it should be noted that there is always the chance that high correlations can result from unresolved co-variances or from outliers in the data, and thus a validation of the most interesting sensitivities should be obligatory. The validation can be done by running additional experiments and or visualizing the displacement and performance values of individual vertices. As an example, two vertices with the label  $v244522$  and  $v265728$  have been selected from the roof and the side of the reference mesh. For both vertices the change in the drag force  $\phi_{FD}$  over the displacement values are plotted for all designs in the joint dataset number 4, see Figs. 4.5 a) and b)). For vertex  $v244522$  a positive and for vertex  $v265728$  a negative correlation value was derived, what is in line with the scatter plots. For completeness, the color of each scatter shows the respective rear lift variation.



**Figure 4.5:** Visualization of the drag change over the displacement values of two selected vertices  $v_{244522}$  and  $v_{265728}$  with respect to the joint dataset 4. The scatter plots allow a qualitative validation of the calculated correlation values.

### 4.4.2 KNN Sensitivities

Global sensitivity methods take all designs in a dataset into account to evaluate the correlation of a vertex to performance changes. In order to control the locality of the calculated sensitivity estimates  $K$ -nearest neighbor algorithms can be applied. The algorithm allows the selection of a reduced set of  $K$  neighboring designs from the entire design dataset, which are closest to the chosen reference design. The automatic identification of the  $K$  neighbors requires the definition of an overall shape distance. Therefore, in this work the absolute values of all vertex displacements between two surface meshes are integrated as follows:

$$d(\mathcal{M}_r, \mathcal{M}_m) = Q_{75}(|\Delta_{i,j}^{r,m}|), \quad (4.20)$$

where  $\mathcal{M}_r$  and  $\mathcal{M}_m$  are the reference and the compared surface,  $n$  refers to the number of vertices  $\vec{v}_i$  on the reference design and the index  $j$  defines the corresponding vertex on the compared mesh. In order to reduce the wait on excessive local deformations, the 75% quantile  $Q_{75}$  is taken instead of the maximum value or the sum of the displacements. Based on the distance measure the  $K$  closest designs are selected a priori to the calculation of the sensitivities.

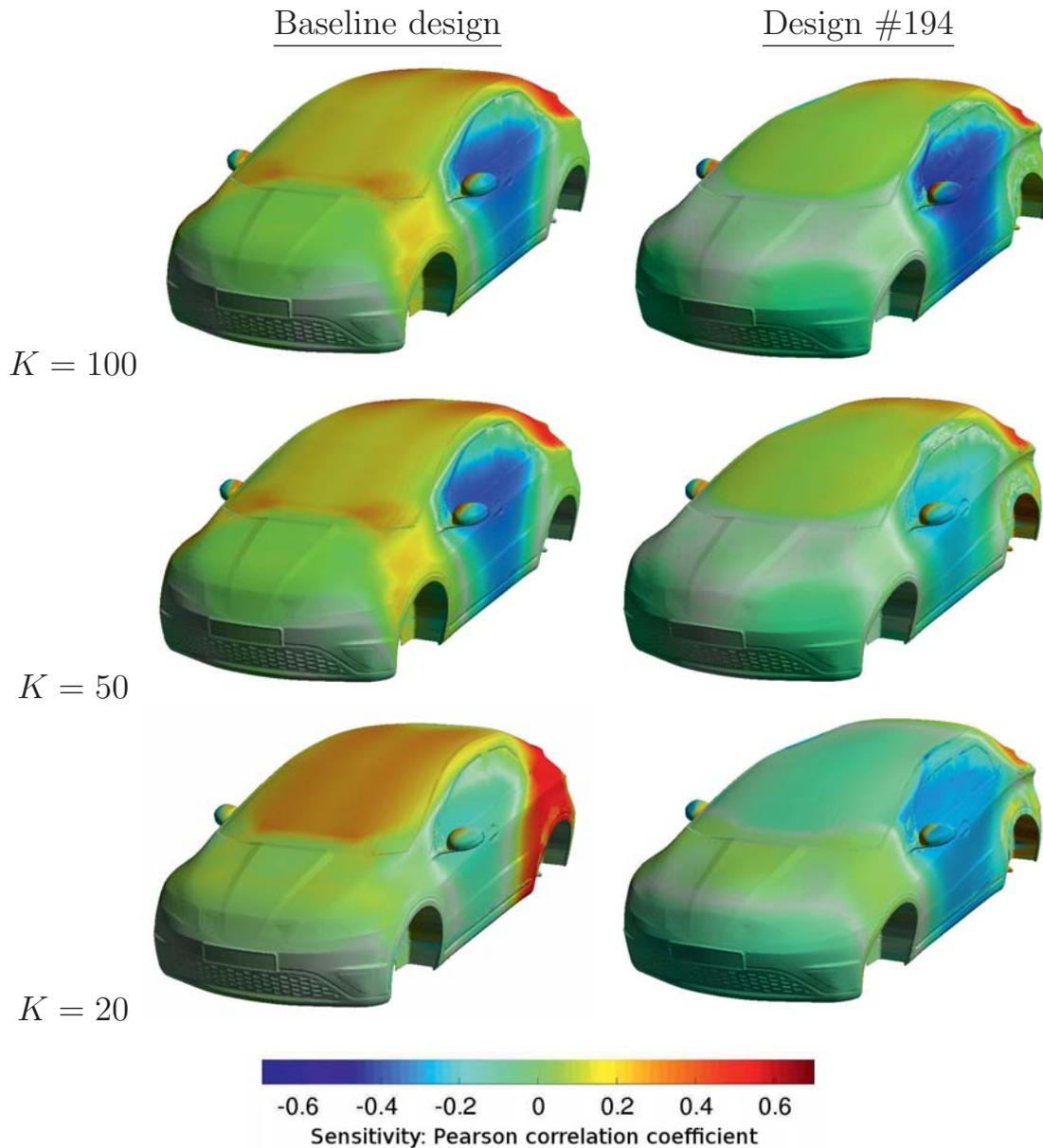
Given dataset 1, where sampling techniques have been applied to generate the design data using representation  $\mathcal{R}_A$ , the KNN based sensitivities have been calculated for  $K \in 20, 50, 100$ . The results are depicted in Fig. 4.6. There, the sensitivities to  $\phi_{FD}$  (drag force) are visualized. The results depict two different aspects. First, the results show how the estimated sensitivities change with different numbers of  $K$ , and second they show how the sensitivities change for different reference designs.

In the left column of Fig. 4.6 the sensitivities with respect to the baseline shape are visualized. For large number of  $K$  with  $K = 100$ , the sensitivity around the side and rear of the car mostly agree with the results of the global sensitivity analysis shown in Fig. 4.4 a). Limiting the sensitivity estimation to the  $k = 20$  closest designs has result in vanishing sensitivities at the side of the car around the front door, and new positive sensitivities have emerged at the side close around the rear wheel.

In the right column of Fig. 4.6 the sensitivities with respect to a different baseline shape are visualized. Comparing the sensitivities of shape #194 to the baseline shape sensitivities for large  $K$ , with  $K = 100$ , reveals a high correlation except for areas around the front screen. Reducing the number of  $K$  results in more distinct sensitive areas. For smaller values of  $K$  the likelihood is reduced that the same designs are used for the calculation of the sensitivities at the position of different reference designs.

In general, the choice of the reference design and the number of  $K$  allows to control where and in which sup-area of the design space the sensitivities are calculated. It should be noted, that the sensitivities can only be expected to be valid for the considered sub-space. Similarly to the choice of  $K$  a threshold can be applied to the shape distance (equation (4.20)) to select designs within a certain radius around the reference design.

#### 4.4. APPLICATION TO PASSENGER CAR DESIGN



**Figure 4.6:** Visualization of the results calculating KNN Pearson sensitivities for different number of  $K$  neighboring designs. On the left and right side the results of the sensitivities with respect to the baseline and an arbitrarily selected alternative reference design are shown. The results are calculated based on dataset 1.

### 4.4.3 Reliability of Probabilistic Sensitivity Estimates

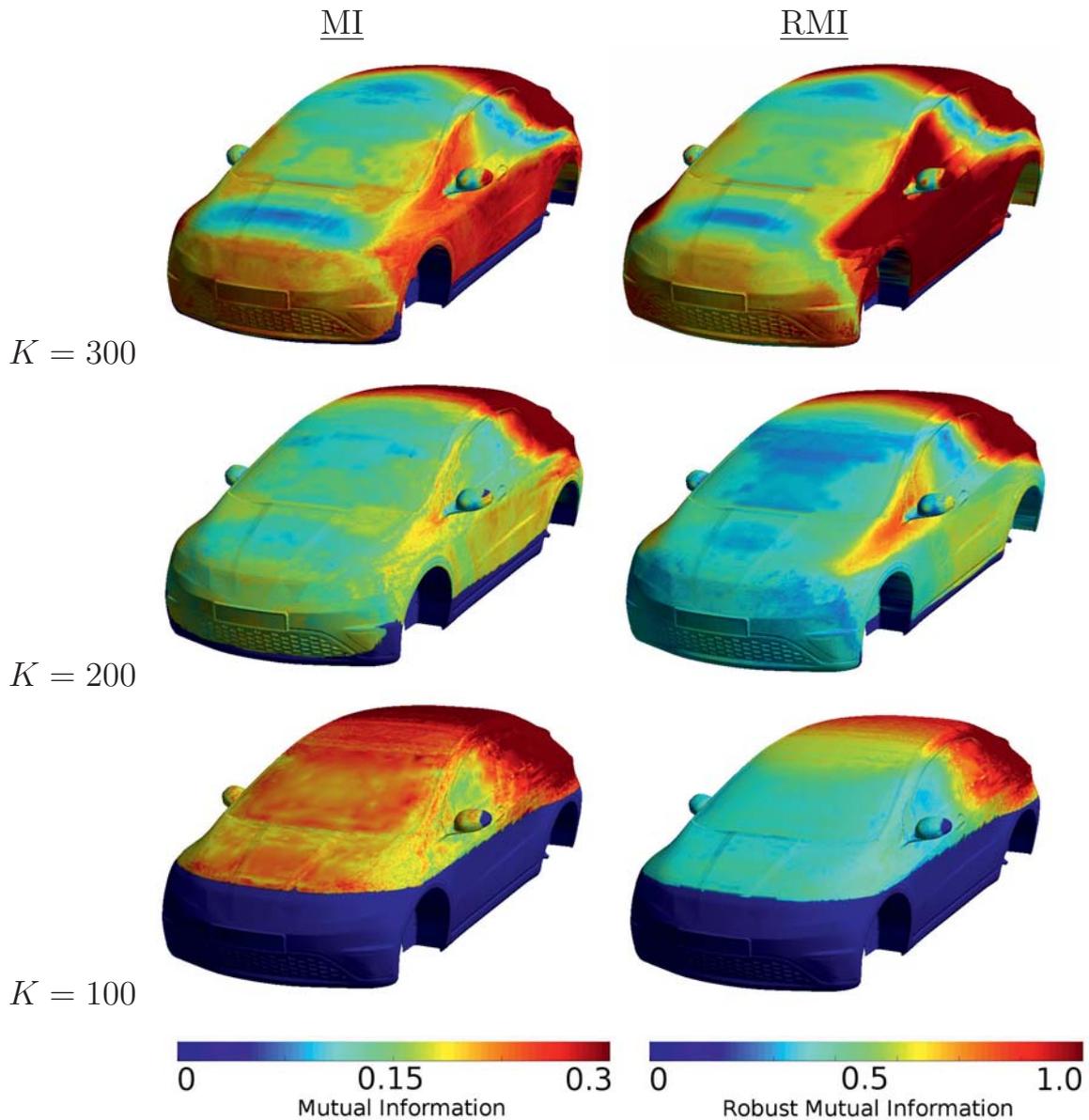
Mutual information (MI) or its robust variant (RMI), see equations (4.6) and (4.16) for reference, provide an alternative and more general attempt for the sensitivity estimation, which compared to the correlation coefficient make no assumption on the kind of interrelation between design modifications and performance changes. However, for a reliable estimation a sufficient number of data samples needs to be available for the analysis. Based on the given joint passenger car design dataset 4 the dependency of the mutual information and the robust mutual information to the size of the design dataset is studied. Using the KNN algorithm, examined in the preceding section, the sensitivities have been calculated for different sample sizes, with  $K \in \{100, 200, 300\}$ . The respective results of the comparison between the mutual information and its robust variant are shown in Fig. 4.7. The sensitivities have been color coded, with the scaling depicted at the bottom of Fig. 4.7. The view on the colored surfaces denotes sensitivities of the vertex displacements to the drag value as follows:

- **Red:** The performance difference  $\phi_{F_D}$  is *sensitive* to deformations of the respective surface patch;
- **Blue:** The performance difference  $\phi_{F_D}$  is *insensitive* to deformations of the respective surface patch.

Hence, the interpretation of the respective sensitivities differs from the interpretation of the Pearson correlations.

For  $K = 300$  out of  $N_D = 336$  designs the sensitivities are comparable to the Pearson correlations in Fig. 4.4 d) (left). There, the MI and RMI reveal all the sensitivities at the side and along the rear of the car shape, which have also been identified using the Pearson correlation coefficient. Beyond that, additional non-linear sensitive regions can be identified at the front of the car, around the area of the number blade.

Comparing the results of the MI and RMI for  $K = 300$  the results of the MI qualitatively coincide with the results of the RMI. Thus, for large number of designs no significant difference between the two sensitivity measures have been observed. But, if  $K$  is reduced, the mutual information fails to clearly separate sensitive from insensitive areas, which gets apparent for  $K = 100$ . There, the sensitivities are calculated based on the RMI and compared to the Pearson correlations, see Fig. 4.4 c) (left). The results show that especially for the analysis of small design datasets the robust mutual information should be preferred to the classical mutual information.



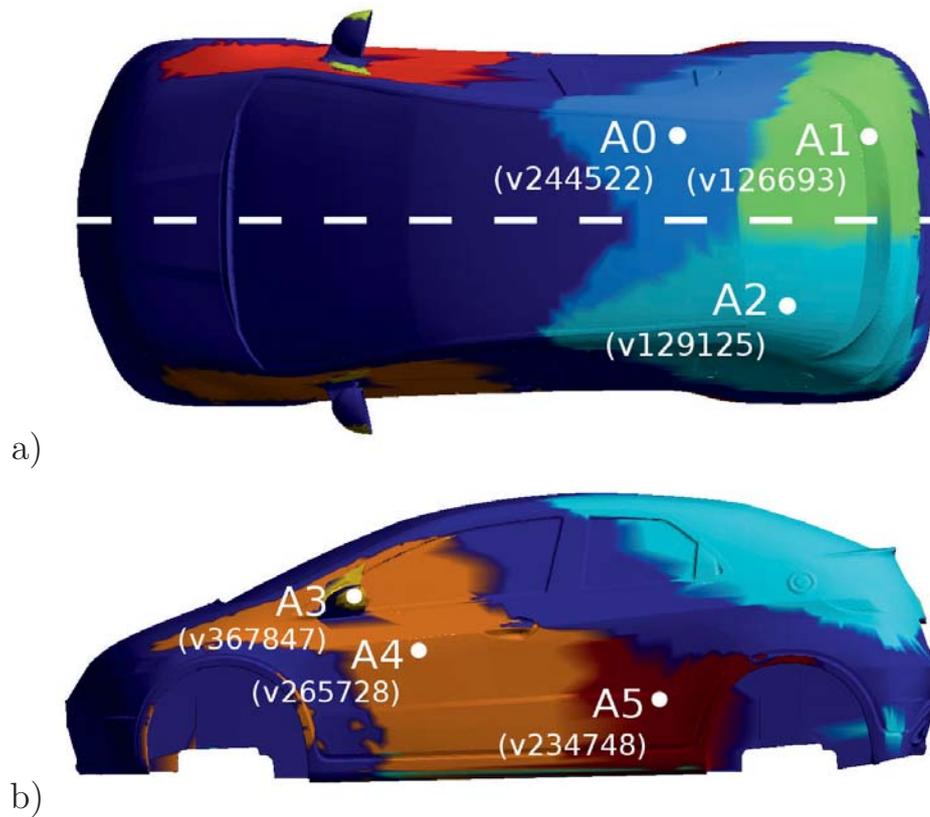
**Figure 4.7:** Comparison of the sensitivity analysis using mutual information (left) and the robust variant of the mutual information (right), calculated based on the data of the  $K$  designs closest to the reference design. The results depict the dependency of the sensitivity measures to the number of designs.

#### 4.4.4 Sensitive Areas

The number of vertices that defines the shape of each design in the datasets is around  $n = 550000$ . A reliable modeling of the interrelations between distant design regions based on all vertices is hardly possible. Therefore, the number of considered vertices and feature variations has been reduced prior to any modeling step. The method for the identification of sensitive areas (Section 4.2) serve as tools for feature reduction. In order to form sensitive areas along the baseline passenger car surface the sensitivity information from the Pearson correlation analysis, see Fig. 4.4 d) and Fig. A.1, has been used. The sensitivities reflect the correlation between the vertex displacements and the change in the drag force with respect to the combined optimization dataset. Together with the geodesic distance, the sensitivity values define the similarity between vertices according to equation (4.19).

In a pre-processing step the feature data of vertices with a low sensitivity,  $|r_{\Delta_i}^r| < 0.6$ , has been removed before the clustering step. Thereafter, the K-Means clustering algorithm is used for an automatic formation of sub-areas on the surface. The results of the clustering are illustrated in Figs. 4.8 a) and b), where vertices with the same color value build one cluster, except for the removed low sensitive vertices which have been colored in dark blue. With  $k = 12$ , the number of clusters has been predefined. Based on the resulting clusters the vertices closest to their centers define the cluster centers.

As can be seen from the results, the sensitive areas are not symmetric along the center line, while the surface mesh is. Such symmetry requirements need to be explicitly incorporated into the clustering process. This has not been considered so far and remains future work. For simplicity, the final selection of the cluster centers has been corrected manually. Considering the symmetry of the passenger car,  $k = 6$  cluster centers have been selected from the entire set of vertices. The selected centers are labeled in Fig. 4.8. The displacement values related to the extracted cluster centers are the basis for the identification of car concepts and the investigation of higher order interaction patterns between distant design areas.



**Figure 4.8:** Results of the feature reduction step by means of identifying sensitive areas. The derived cluster centers and the corresponding feature difference values define an abstract representation of the individual surface areas.



### 4.5 Interim Summary

---

In this chapter different methods for the estimation of vertex sensitivities have been studied. The sensitivities denote information about the interrelation between vertex feature disparities like the displacements and variations of the design performance. The visualization of the sensitivities on the surface mesh provide the engineers with prospective effects on the performance indicators with respect to possible shape deformations.

Using linear correlation techniques like the Pearson correlation coefficient, not only information about the sensitivity but also about the expected deformation direction for the improvement of the performance can be estimated. It has been demonstrated that the application of K-nearest neighbor algorithms a priori to the sensitivity calculation let the engineer control the locality of the sensitivity estimates.

The application of the mutual information and its robust variant for sensitivity analysis provide a more general attempt to the evaluation of vertex sensitivities. From the results on the passenger car design data, it can be concluded that especially for a small number of designs the robust variant of the mutual information is beneficial over the standard mutual information. For large design datasets both measures have provided comparable results.

An algorithm, which exploits the sensitivity measure and the Geodesic distance between vertices, has been introduced to derive larger sensitive areas on the design surface. Even though the surface differences considered in our research are not huge, it has been shown that the Geodesic distance is more suitable than the standard Euclidean distance. The resulting sensitive areas and the related design features make up a low dimensional representation of the design data. As such, it facilitates the modeling of the interrelations between distant design regions and constitutes an important step in the overall shape mining process.



# 5

## Design Concept Retrieval

Feature reduction by the identification of sensitive areas, introduced in the preceding Chapter 4, facilitates an enhanced modeling of the design data. In this chapter, modeling techniques for the retrieval, representation and evaluation of abstract *design concepts* are described. After a general definition of the term design concept, formulations from association rule learning have been adopted to represent concepts in human readable form, which is desired to simplify the knowledge transfer to the design engineers. Subsequently, methods for the evaluation of design concepts are reviewed and a new *measure of utility* is presented, which allows to rank design concepts according to the formulation of the engineers' objectives. Exemplary, two modeling techniques from machine learning, decision trees and self-organizing maps (SOMs), are studied for an automatic retrieval of concepts from the design data. The generalized process for the retrieval, representation and evaluation of design concepts is applied to the passenger car design data to emphasize the usability in real world design scenarios.

### 5.1 Definition

---

The term *Design Concept* (and what makes up such a design concept) lacks a clear definition in the literature. Towards a working definition, the term *Design* is reviewed based on [Ralph and Wand, 2009]. The authors understand a design as: "*a specification of an object, manifested by an agent, intended to accomplish goals, in a particular environment, using a set of primitive components, satisfying a set of requirements subject to constraints*". Following this formulation, it is essential to note that a design is not a object specification solely. Rather, a design is a specification of an object that has been created to accomplish a pre-defined goal. The compliance with the design goal is typically evaluated based on observed functional or aesthetic properties of the design, which result from its interaction with the environment.



## CHAPTER 5. DESIGN CONCEPT RETRIEVAL

---

To denote how design concepts can be derived from a set of designs, the notion of a *Concept* needs to be discussed. Various formulations of a concept or conception have been published by several philosophers. Most of them agree on the fact that a concept is defined based on the abstraction from observations of the world, e.g., in [Mill, 1882], J. S. Mill wrote: “[...] and when we form a set of phenomena into a class, that is, when we compare them with one another to ascertain in what they agree, some general conception is implied in this mental operation.”

Concepts itself and the process of formulating concepts is treated as a mental process throughout the definitions in the literature. They are mentally processed ideas or abstractions, representing summaries of properties from individual observations. Once learned, utilizing concepts allows to structure the world, where concepts function as filters for future perceptions. Furthermore, concepts are of use if they allow to explain goal relevant portions of the world. As an example, if survival is seen as the goal of human beings, the identification of concepts for eatable and uneatable food is of high relevance. Thus, useful concepts support humans to make the right decisions upon future observations.

Consolidating the key aspects of the notion of designs and concepts, led us to the following definition of a design concept:

**Design Concept** A design concept is an abstract representation of a set of designs sharing an akin specification that map onto approximately equivalent design properties which are characterizing the design goal.

The identification and representation of design concepts intents to structure the design domain by generalizing from individual design solutions. Design concepts allow the classification of individual designs based on object and functional properties. In the light of computational intelligence, the process of concept learning and representation is transferred to computer systems, admitting that concept formation is not only subject to living creatures but is an ability of intelligent systems in general.

Discovering a discrete and compact set of design concepts has the potential to support the engineers in extracting their own concepts and allow a purposive decision making within the design process. Before that, an automatic extraction, description and evaluation of design concepts from design data is required. Modern technologies from data mining and artificial intelligence provide instruments to achieve this task.

## 5.2 Concept Representation

---

A compact and human readable representation of design concepts is desired to simplify the understanding and processing of the domain information by design engineers. In the field of machine learning and data mining, summaries and abstractions from data are commonly represented by rules of the form,

$$IF \text{ [antecedent]} \text{ THEN [consequent]},$$

or using a more compact notation,

$$\{\text{antecedent}\} \rightarrow \{\text{consequent}\},$$

where the antecedent and consequent part of a rule are defined by a logical concatenation of attribute-value relations. For a general introduction into *rule induction* and *association rule learning*, see [Kantardzic, 2001, chap. 13 and 16]. When adopting the formulation of rules for the description of design concepts, the antecedent relates to an abstract object specification and the consequent part to functional or other goal relevant properties of the designs.

Depending on the used attributes the rules can be classified into qualitative and quantitative rules. Qualitative rules refer to discrete states of the attributes while quantitative rules refer to real-valued features. Specific to a dataset  $\mathbf{D}^r = \{\Delta^r, \phi^r\}$ , which comprises data about the shape differences  $\Delta^r$  and the performance differences  $\phi^r$ , the rules refer to a pre-defined reference design  $r$ . Rules that describe the cause and consequence of an association relative to a design are denoted as *relative design rules*. Thus, for a correct interpretation all rules have to be related to the specification of the reference design. Related to  $\mathbf{D}^r$ , a qualitative description of the design concepts can refer to the sign of the feature and performance differences. Because those rules are expected to describe a wider area of the design space, such rules often describe more general concepts. *Qualitative rules* representing design concepts can formally be defined by the following generic association:

$$\bigwedge_i (\Delta_i^r \{<, \geq\} 0) \rightarrow \bigwedge_j (\phi_j^r \{<, \geq\} 0). \quad (5.1)$$

In order to explain this definition, let's consider the following example of a qualitative rule:

$$(\Delta_1^0 > 0) \wedge (\Delta_2^0 < 0) \rightarrow (\phi_1^0 < 0).$$

This rule can be interpreted as follows, an increase of the feature value  $\Delta_1^0$  of  $\vec{v}_1$  relative to the reference design  $r = 0$ , together with a decrease in the feature value  $\Delta_2^0$  of  $\vec{v}_2$  results in a decrease of the performance  $\phi_1$ .

Quantitative association rules, see [Miller and Yang, 1997; Srikant and Agrawal, 1996], consider the feature values explicitly as real-valued attributes. There, the antecedent and consequent of a rule are composed of attribute interval relations. A design is covered by a quantitative rule if its attribute values are within a defined range. With respect to  $\mathbf{D}^r$ , *quantitative rules* are defined in the following form:

$$\bigwedge_i (\Delta_i^r \in [l_i, r_i]) \rightarrow \bigwedge_j (\phi_j^r \in [l_j, r_j]). \quad (5.2)$$

In contrast to qualitative rules, quantitative rule types can describe more specific concepts, where the antecedent describe more local areas in the design space. The following association depicts an example for a quantitative rule:

$$(\Delta_1^0 \in [0.2, 0.5]) \wedge (\Delta_2^0 \in [-1.0, 0.0]) \rightarrow (\phi_1^0 \in [10, 20]).$$

This rule can be interpreted as follows: if the feature difference of  $\vec{v}_1$  takes values between 0.2 and 0.5, and if the feature difference of  $\vec{v}_2$  is between  $-1.0$  and  $0.0$ , the performance difference to the reference design  $r = 0$  is expected to be between 10 and 20.

### 5.3 Concept Evaluation

---

The computational retrieval of design concepts can result in a large set of concepts, which can hardly be analyzed at once by engineers or data mining experts. The investigation of a huge number of concepts takes considerable time and requires huge amount of human resources. An a priori evaluation of individual design concepts allows the ordering and filtering based on their relevance. In such a way, potentially uninteresting concepts can be hidden from the engineer's eye. To identify an adequate evaluation measure, commonly used approaches from association rule learning and analysis are reviewed, see [Webb and Zhang, 2002; Tan et al., 2004; Geng and Hamilton, 2006; Webb, 2006] for more details. Considering the definition of engineering objectives, a new measure of *utility* is suggested to order design concepts in the domain of engineering design.

### 5.3.1 Relevance and Interestingness

Measures of relevance or interestingness can be classified into *objective* and *subjective* indicators. While objective measures rely on the statistics of the raw data only, subjective measures take additionally or solely engineer's experiences and preferences into account. In general, objective measures target the quantification of concepts like *generality*, *specificity* or *accuracy*, while measures of *interestingness*, *novelty* and *surprisingness* belong to the class of subjective measures.

For the following explanations we consider that design concepts are represented by a general form of an association rule  $A \rightarrow C$  with antecedent  $A$  and consequent  $C$ . Further, it is considered that a design dataset  $\mathbf{D}$  with  $N$  designs is given, against which the concepts are evaluated.

#### Comprehensiveness

In association rule learning, two indicators, the *coverage* and the *support* are used to measure the comprehensiveness of a design concept. A design is covered by a concept if and only if every condition in the antecedent part of the rule is satisfied. In probabilistic terms, the coverage of the design concept is defined as the probability  $P(A)$  that a design meets all conditions in  $A$  given the designs in  $\mathbf{D}$ :

$$\text{cov}(A \rightarrow C, \mathbf{D}) = P(A). \quad (5.3)$$

If the design is covered by a concept and in addition fulfills all conditions in the consequent part, such design is said to *support* the considered design concept. The measure of support over all designs in  $\mathbf{D}$  is defined by the joint probability that a design meets all conditions in  $A$  and  $C$ :

$$\text{supp}(A \rightarrow C, \mathbf{D}) = P(AC). \quad (5.4)$$

The support provides information about the generality of a design concept with respect to the considered design dataset. Concepts with low support are often too specific to be of major interest. However, whether a concept with high support is more interesting to the engineer depends on his or her experiences. Concepts with high support are likely to represent trivial associations, which are already known to the expert.

#### Reliability

If the *confidence* is high that the ascribed object specification  $A$  causes the abstracted consequent  $C$ , such an association is called reliable. The measure of confidence can be defined in terms of coverage and support as follows:

$$\text{conf}(A \rightarrow C, \mathbf{D}) = \frac{\text{supp}(A \rightarrow C, \mathbf{D})}{\text{cov}(A \rightarrow C, \mathbf{D})} = P(C|A) = \frac{P(AC)}{P(A)}. \quad (5.5)$$

If the concept covers many designs but has a low support, the confidence with respect to the dataset  $\mathbf{D}$  is low. Design concepts with low confidence are potentially uninteresting to the engineer and can often be rejected from further considerations. However, using the confidence measure only to quantify the relevance of an association can result in misleading evaluations, see [Brin et al., 1997; Silverstein et al., 1998; Tan and Kumar, 2000].

Under the assumption that the strength of the correlation between antecedent and consequences reflects the *interestingness* of an association, in [Tan and Kumar, 2000], the authors introduced a new measure that takes the confidence of the association and its reversal into account. The metric is defined as follows,

$$\text{IS}(A \rightarrow C, \mathbf{D}) = \sqrt{\text{conf}(A \rightarrow C, \mathbf{D}) \times \text{conf}(C \rightarrow A, \mathbf{D})}. \quad (5.6)$$

Equivalent to the formulation in terms of confidence, the measure of interestingness can be defined in terms of marginal and joint probabilities:

$$\text{IS}(A \rightarrow C, \mathbf{D}) = \sqrt{\frac{P(AC)}{P(A)P(C)} P(AC)}, \quad (5.7)$$

where  $P(AC)$  reflects the support of the association and  $P(AC)/(P(A) \cdot P(C))$  quantifies the dependency between antecedent and consequence. *IS* has shown to allow a sensible ordering of the associations according to the interesting assumption made. Although, the authors term their measure an measure of interestingness, subjective ratings or preference of the concepts are not included.

### 5.3.2 Measure of Utility

While measures like support or confidence are objective measures and do not take any preference of the engineers into account, subjective measures of *interestingness* require a direct interaction with a human evaluator. However, formulating distinct assumptions about the engineer's background and intention can facilitate the definition of an objective quantification of interestingness. Thus, preferences of the evaluator are formulated a priori to the evaluation of concepts. Definitions of *correlation*, *peculiarity*, *utility*, *novelty* or *applicability* can be utilized to define particular measures of interestingness, see [Geng and Hamilton, 2006].

In the following a new measure of *utility* is introduced, which explicitly takes information about the engineer's pre-defined design objectives into account. While measures of *peculiarity* and *novelty* can be defined based on a general level of abstraction, measures for *utility* and *applicability* are problem dependent. For example, a design concept under evaluation could be judged as peculiar if its ascribed functional properties differ significantly from other concepts which cover designs with similar design specifications. Whether such a peculiar concept is novel or not depends on the concepts that have been retrieved in the past. For example, in [Graening et al., 2010], novelty has been evaluated against an approximation model abstracted from past solutions. In engineering design the *applicability* of a solution or concept is most often linked to questions of manufacturing, material or other costs, typically given as constraints in the design process. Already the direct analysis of the design properties with respect to design constraints can lead to a proper measure for applicability. For example, if the minimum thickness of a geometry is below a certain threshold the final object might be impossible to manufacture. When labeling such infeasible or expensive designs in the data set, measures of support and confidence could already be applied for quantifying the applicability of a design concept.

A concept that is not applicable is not necessarily of low *utility*. First, a concept can be of high utility if it adds new relevant information to a knowledge base and, e.g., allows a more accurate estimation of novelty and peculiarity. Second, a design component might be of high utility if it improves engineer's understanding and helps to reach the design goal. The objective values that quantifying the design goal are typically well defined allowing to derive a measure of utility.

The objectives denoted as  $O$  are typically formulated with respect to the design specification and functional properties of the design, so that  $A \wedge C \rightarrow O$ . In the case that  $O$  contains multiple quantitative attributes, performance indicators used in multi-Objective optimization, see [Deb, 2001], can be adopted to evaluate the compliance of a concept with the objectives. One such measure is the hypervolume indicator defined in [While et al., 2012]. Following [While et al., 2012]: "The hypervolume of a set of solutions measures the size of the portion of objective space that is dominated by those solutions collectively." According to [Deb, 2001] a solution  $a$  is said to dominate a solution  $b$  if for each objective, solution  $a$  equals or outperforms solution  $b$  and solution  $a$  at least outperforms  $b$  with respect to one objective.

Based on the hypervolume indicator and the  $IS$  measure a new measure of relevance is defined evaluating the utility of a design concept:

$$\text{util}(A \rightarrow C, \mathbf{D}) = \text{vol}(O, \mathbf{D}) \cdot \text{IS}(A \rightarrow C, \mathbf{D}) \quad (5.8)$$

where  $\text{vol}(O, \mathbf{D})$  is the hypervolume calculated based on the non-dominated solutions of all designs covered by the concept  $A \rightarrow C$ . A comparison of different methods for the hypervolume computation has been carried out by [Priester et al., 2013]. In general, for the calculation of the hypervolume a reference point has to be defined, typically referring to the worst solution in the objective space.

As a consequence of the combination of the  $IS$  measure and the hypervolume indicator, a design concept is of high utility if it has a high confidence that the association described by the concept is true for all designs and that it manages to achieve the objectives defined by the design engineers best.

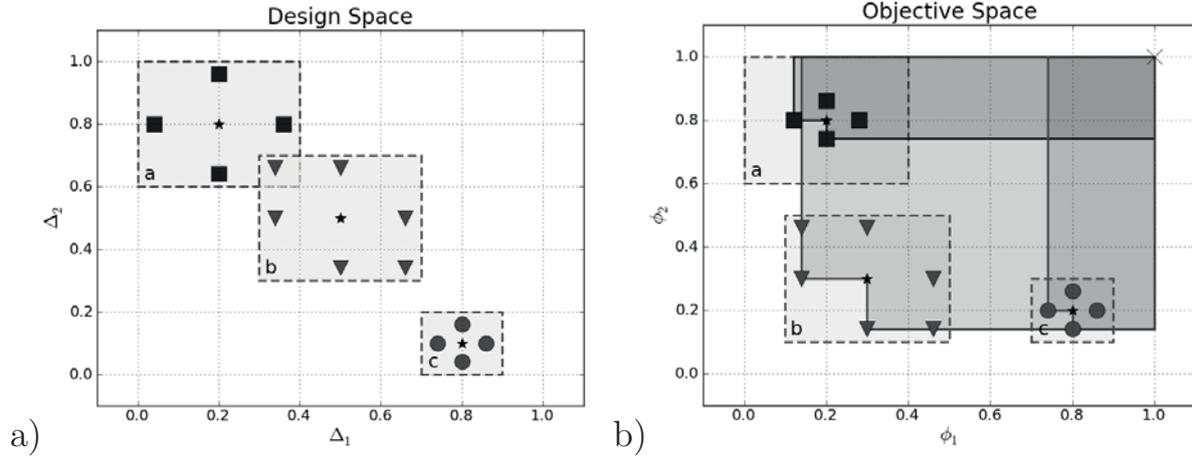
### Example

A simple illustrative example should make the difference between the different evaluation measures apparent. In this example three concepts  $a$ ,  $b$  and  $c$  are represented by the following quantitative design rules:

$$\begin{aligned} a : \Delta_1 \in [0.0, 0.4] \wedge \Delta_2 \in [0.6, 1.0] &\rightarrow \phi_1 \in [0.0, 0.4] \wedge \phi_2 \in [0.6, 0.9], \\ b : \Delta_1 \in [0.3, 0.7] \wedge \Delta_2 \in [0.3, 0.7] &\rightarrow \phi_1 \in [0.1, 0.5] \wedge \phi_2 \in [0.1, 0.5], \\ c : \Delta_1 \in [0.7, 0.9] \wedge \Delta_2 \in [0.0, 0.2] &\rightarrow \phi_1 \in [0.7, 0.9] \wedge \phi_2 \in [0.1, 0.3]. \end{aligned}$$

The concepts are evaluated against a set of solutions  $\mathbf{D}$  containing  $N = 14$  designs. Figs. 5.1 depict the solutions and concepts in the design space a) and in the objective space b). The objective values are defined based on the performance differences,  $\phi_1$  and  $\phi_2$ , with the objective to minimize both values.

For each concept the coverage, support, confidence,  $IS$ , hypervolume and utility are calculated. The results are summarized in Table 5.1. The concepts are ordered according to the measure of utility. Concept  $b$  covers the largest proportion of the design solutions with respect to their specification in the design space. Hence, the highest coverage value is assigned to concept  $b$ , followed by concept  $a$  and  $c$ . If the covered solutions are all associated with the correspondent consequence then  $\text{supp}(A \rightarrow C, \mathbf{D}) = \text{cov}(A \rightarrow C, \mathbf{D})$  and the confidence reaches its maximum with  $\text{conf}(A \rightarrow C, \mathbf{D}) = 1$ , as can be seen from the evaluation results of concept  $b$  and  $c$  shown in Table 5.1. But, if solutions are covered by the left side of the rule which do not match the expected consequence, then  $\text{supp}(A \rightarrow C, \mathbf{D}) < \text{cov}(A \rightarrow C, \mathbf{D})$  and as a result the confidence value deviates from its maximum with  $\text{conf}(A \rightarrow C, \mathbf{D}) < 1$ . This can be observed when analyzing the results of concept  $a$ . The confidence of the reverse associations  $\text{conf}(C \rightarrow A, \mathbf{D}) = 1$  for all



**Figure 5.1:** Illustration of the example design dataset a) in the design space and b) in the objective space. The designs are assigned to three distinct design concepts a, b and c. For each concept the area which makes up the hypervolume is denoted in the objective space b).

concepts. Thus, the  $IS$  measure equals 1.0 for concept  $b$  and  $c$ . For concept  $a$ , where  $\text{conf}(A \rightarrow C, \mathbf{D}) = 0.801$ ,  $IS(A \rightarrow C, \mathbf{D}) < 1.0$ , indicating a non-perfect correlation between the antecedent and the consequence.

Concept	cov	supp	conf	IS	vol	util
$b$	0.429	0.429	1.000	1.000	0.714	<b>0.714</b>
$c$	0.286	0.286	1.000	1.000	0.220	<b>0.220</b>
$a$	0.357	0.286	0.801	0.895	0.224	<b>0.200</b>

**Table 5.1:** Evaluation results from the exemplary concept candidates (Figs. 5.1) ordered according to their utility value.

The hypervolume for all three concepts has been calculated with respect to a pre-defined reference point at  $(1.0, 1.0)^T$  in the objective space. The non-dominated solutions within concept  $b$  span the largest hyper volume. Together with an  $IS$  value of  $IS = 1.0$ , concept  $b$  is evaluated to be of highest utility. The calculated hypervolume for the remaining two concepts  $a$  and  $c$  are similar. But, since the  $IS$  value of  $a$  is smaller compared to  $c$ , concept  $c$  is of higher utility. It shows that concepts with a large hypervolume not necessarily have to be of high utility.



### 5.4 Concept Retrieval

---

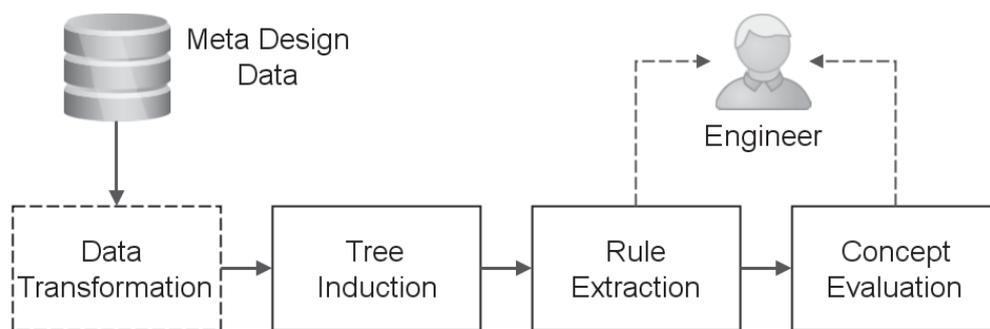
The automatic retrieval of design concepts from a set of designs, including their performance indicators, is directly related to methods for concept learning and classification, see [Mitchell, 1997] and [Duda et al., 2001], respectively. Classification algorithms can be adopted to automatically derive abstract representations of the designs based on their design features and performance indicators. Therefore, a model of distinct classes is learned from the training dataset that minimizes the classification error, e.g., the misclassification rate, [Kantardzic, 2001, chap. 6]. The training of the classifier is related to the identification of boundaries that partition solutions in the design space based on their similarity. The resulting model maps different abstract design specifications to different class labels, e.g., representing performance numbers.

In machine learning and data mining a large amount of classification algorithms have been studied that differ mostly in the way the classification boundaries are constructed. Among the most prominent ones are Decision Trees [Breiman et al., 1984], Rough Set Theory [Pawlak, 1982, 1991], Fuzzy Sets [Zadeh et al., 1996] and Artificial Neural Networks [Zhang, 2000]. The choice of the classifier should be based on the characteristics of the design data, the desired classification error as well as on the possibilities to represent the retrieved concepts in a human readable way. Two classification algorithms, the classification and regression tree (CART) and the Self-organizing map (SOM) are introduced in more detail. Those algorithms are used for the induction of design concepts in subsequent experimental studies.

#### 5.4.1 Decision Tree

Decision trees are supervised learning models frequently used in data mining, machine learning and other domains. Their popularity comes from their conceptual simplicity, from their interpretable structure and because they can be applied for regression and classification tasks similarly, see [Breiman et al., 1984; Safavian and Landgrebe, 1991; Rokach and Oded, 2005]. Decision trees are a group of classification models that are constructed by recursively splitting the input space into hyper-rectangular sub-spaces. They are represented by a directed graph that consists of a finite set of nodes and branches connecting them. One distinguishes between the root node, internal or test nodes and terminal nodes representing the leaves of the tree. The root and internal nodes represent attributes at which conditions are tested and solution sets are split into two or more homogeneous sub-sets. The branches represent the outcome of the split routing to a sub-area of the input space. A class

or target value is assigned to each node abstracting the label information of all solutions in the represented sub-space. In contrast to the organic tree, a decision tree is typically grown from top to bottom starting with an individual root node at the top that is representing the entire training set. In each internal node the variable and split-point is chosen based on a quality measure, minimizing the impurity at each node. The impurity criteria controls the growth of the tree and is for classification trees typically defined by either of the three criteria: the misclassification error, the gini index or the cross-entropy, defined in [Rokach and Oded, 2005]. For regression trees the squared regression error or an analogue measure is used. The growth of the tree terminates if an overall quality criteria, e.g., related to the regression or classification performance is met or if a minimum node size is reached. Pruning strategies are applicable for a subsequent reduction of the complexity of large decision trees, see [Safavian and Landgrebe, 1991].



**Figure 5.2:** *Illustration of the process for an automatic extraction of design concepts using decision tree induction. The extraction of qualitative design concepts requires the transformation of the data into discrete states.*

For continuous target variables the induction of decision trees can be carried out in three different ways: discretizing the target variables a priori to the tree induction, applying regression tree learning directly to the continuous variables [Breiman et al., 1984] or dynamically discretizing the target variable while growing the tree [Hu et al., 2009]. In this work, two strategies are considered in order to derive qualitative and quantitative design rules from the tree. Given the design feature and performance differences, the variables are discretized into binary states depending on the sign of the parameter values before the tree induction process. The sign provides information about the increase or decrease of the parameter values with respect to the reference design. The use of the transformed data for tree learning supports the generation of qualitative design rules. In order to derive quantitative rules the regression tree algorithm is directly applied to the continuous

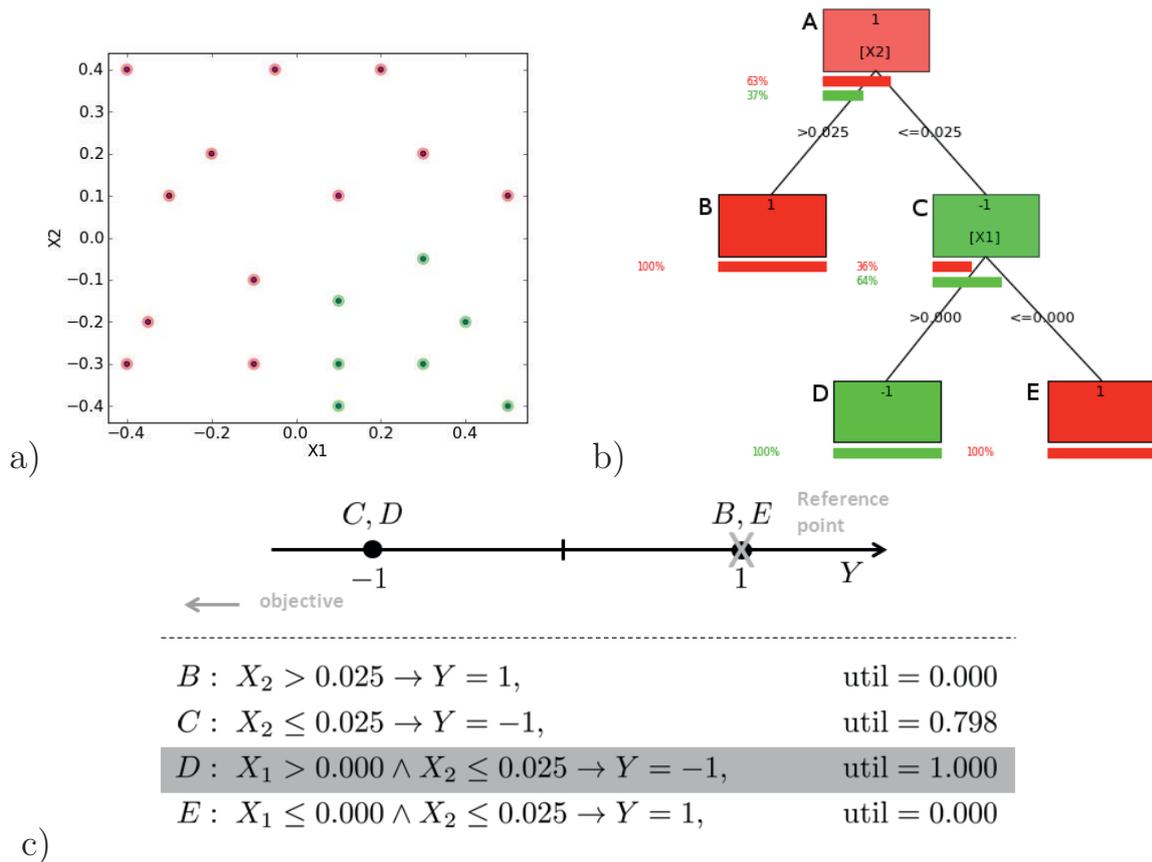
input and target variables. The rules are extracted subsequent to the induction of the decision tree by processing all branches of the tree. The nodes and branches along the path to each node define the antecedent and the statistics of the performance values in the target node define the consequence of each rule. The resulting concepts, represented by the rules, can be evaluated with respect to their utility to emphasize the most relevant concepts.

Fig. 5.2 summarizes the entire process for concept retrieval, rule extraction and evaluation in the context of decision tree learning. As indicated in the graph, the meta design data might need some transformation first. This depends on the desired concept representation.

### Example

The following example illustrates the results of the automatic concept retrieval process using the C4.5 decision tree algorithm. An artificial training dataset containing values for two independent input variables  $X_1, X_2 \in \mathbb{R}$  and a discrete target variable  $Y \in \{-1, 1\}$  has been generated. All data samples of the dataset are shown in Fig. 5.3 a). Solutions that map to the target value  $Y = 1$  and  $Y = -1$  are brushed in red and green respectively. The dataset has been used to derive a binary decision tree. The resulting tree is depicted in Fig. 5.3 b). The *gain ratio* [Quinlan, 1992] has been applied to select the split variable and split point in each node. For simplicity, the growth of the tree has been constrained by a maximum tree depth of three and by forcing a minimum number of three examples that are covered by each node. The nodes  $A$  to  $E$  are labeled with the expected target value at the top. Additionally, nodes  $A$  and  $C$  are labeled with the respective split variable  $X_1$  or  $X_2$ . The colored bars below each node show the distribution of the target values with respect to the represented data. The root node  $A$  is representing all data in the dataset. If a solution is drawn randomly from the entire dataset, the expected target value would be  $Y = 1$ . The decision tree algorithm identified  $X_2$  as the variable which best splits the entire dataset. The edges between the root node and the two child nodes label the split boundary. While  $B$  denotes a terminal node, the internal node  $C$  is splitting again based on  $X_1$  into two subsets separated by the boundary  $X_1 = 0.000$ . The split results in two terminal nodes  $D$  and  $E$ .

The rule based representation of the different concepts are derived by following each branch in the tree. There, each node is representing a potential concept. The antecedent is build based on the different split points and the consequence is the expected value of  $Y$  in each node. Rules are extracted for each node  $B$  to  $E$ . The resulting rule set is shown in Fig. 5.3 c). In addition the utility value is calculated for each concept based on the assumption that



**Figure 5.3:** Illustration of the results from applying decision trees for the extraction and evaluation of design concepts from a simple example data set. The solutions of the example data set are depicted in a), the resulting decision tree graph is shown in b) and the retrieved rules of the concepts and the related utility value are listed in c).

the objective is to minimize  $Y$ , so that  $Y = -1$  is of high and  $Y = 1$  of low quality. The calculation of the hypervolume has been carried out based on the normalized target values with a reference point that was set to  $Y = 1$ . From the evaluation results it can be noted that concept  $D$  is of highest utility with a confidence value of  $\text{conf} = 1.000$  and a hypervolume of  $\text{vol} = 1.000$ . The rule covers all solutions that map to the optimal target value.

### 5.4.2 Self Organizing Maps

Motivated from cortical maps, Self-Organizing Maps (SOM), sometimes referred to as Self-Organizing Feature Maps (SOFM) belong to the class of unsupervised artificial neural networks. SOMs have first been introduced by [Kohonen, 1982] followed by a couple of subsequent publications, e.g., see [Kohonen, 1995a; Kohonen and Honkela, 2007]. Meanwhile Kohonen's

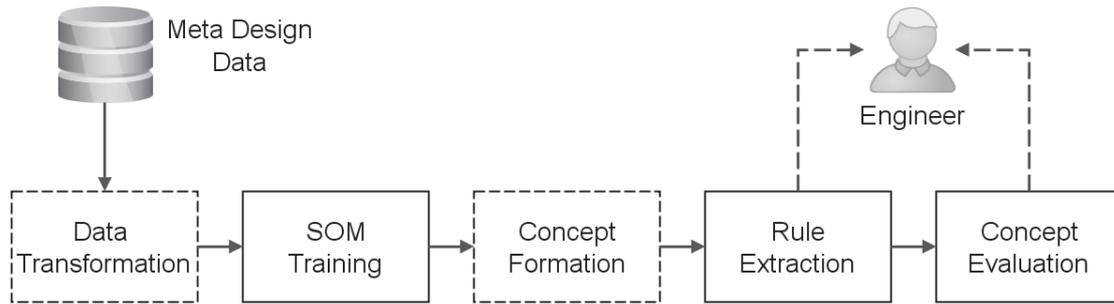
self-organizing maps have been used within a broad variety of applications. A few applications in the engineering domain are summarized in [Kohonen et al., 2002]. For the investigation of structured aerodynamic design data SOMs have first been studied by [Obayashi and Sasaki, 2003]. They have been applied to group and analyze the trade-off of aerodynamic designs with respect to multiple performance criteria.

Given a training data set, SOMs project the high dimensional continuous input space onto a low dimensional discrete output space. It therefore implements a feed-forward network structure consisting of two layers, one input and one output layer referred to as feature map. The structure of the input layer is directly defined by the number of input variables while the topology of the feature map needs to be pre-defined a priori to the training procedure. Typically a 1D or 2D feature map is used where output neurons are organized on a regular lattice. Each neuron of the input layer is fully connected with the neurons of the output layer by continuous weights. The different weights are adjusted during the SOM training to best represent the structure of the input data. After the training phase, each weight vector connected to each output neuron makes up a prototype vector representing a class of similar inputs.

The update of the weight vectors is conducted by iterating the following instructions:

1. *Initialization*: Randomly initialize the weight vectors
2. *Sampling*: At time step  $t$ , draw randomly a sample  $\mathbf{x}(t)$  from the training data.
3. *Competitive learning*: Find the best matching neuron  $\mathbf{m}_c$  of the feature map (winning neuron):  $\mathbf{m}_c = \operatorname{argmin}(d(\mathbf{m}_i, \mathbf{x}))$ , using a pre-defined distance metric  $d$ , e.g., the Euclidean distance.
4. *Cooperative updating*: Update the weights:  $\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(\mathbf{x}(t) - \mathbf{m}_i(t))$ , where  $h_{ci}$  is a kernel function defining the lateral influence on neighboring neurons, the elongation of  $h_{ci}$  is typically adapted over time using a predefined learning rate.
5. *Continuation*: go to step 2 until the weight values reach a steady state.

The training algorithm realizes a topology preserving mapping from the input space to the low dimensional feature map by iteratively applying competitive learning and cooperative updating to the weights. The low dimensional output layer preserves the statistics and structure of the input dataset.



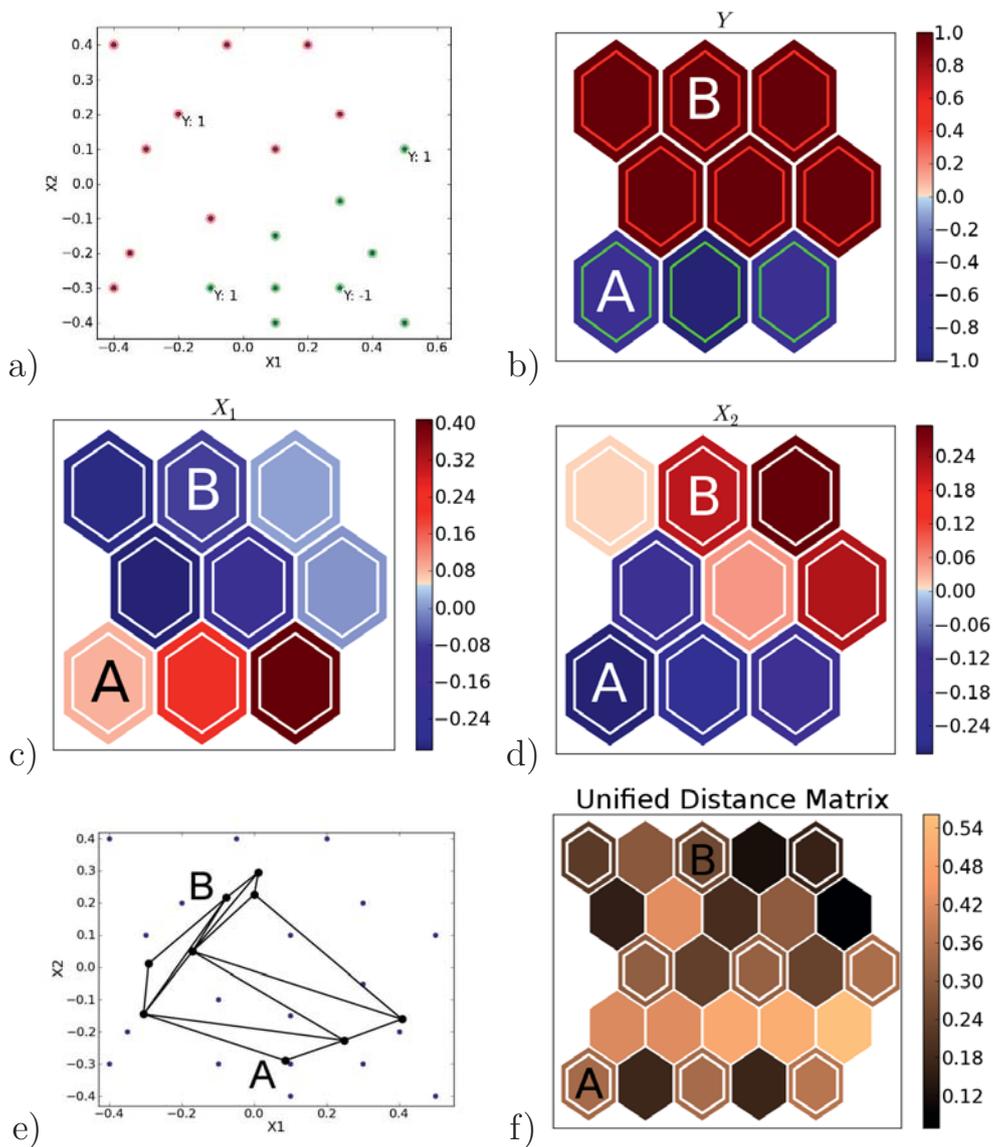
**Figure 5.4:** Illustration of the process for an automatic extraction of design concepts based on connectionist self-organizing maps (SOMs). The formation of concepts and the extraction of rules requires explicit post-processing steps subsequent to the SOM training.

The investigation of the output neurons thus reveals information about the structure of the high dimensional input space.

The original SOM algorithm [Kohonen, 1982] realizes an unsupervised learning procedure without explicitly considering a supervision signal. If a continuous output signal is given, the simplest way to account for a target value is to expand the input vector with the expected output value. However, this might lead to non-optimal mappings especially for high dimensional input data. More elaborate is the application of Learning Vector Quantization (LVQ) algorithms, a supervised variant of SOMs [Kohonen, 1995b; Hammer and Villmann, 2002]. In its simplest form, LVQ first tunes the weight vectors of the network structure using the classical SOM training procedure. In a second step the weights are updated according to a given class label or target value.

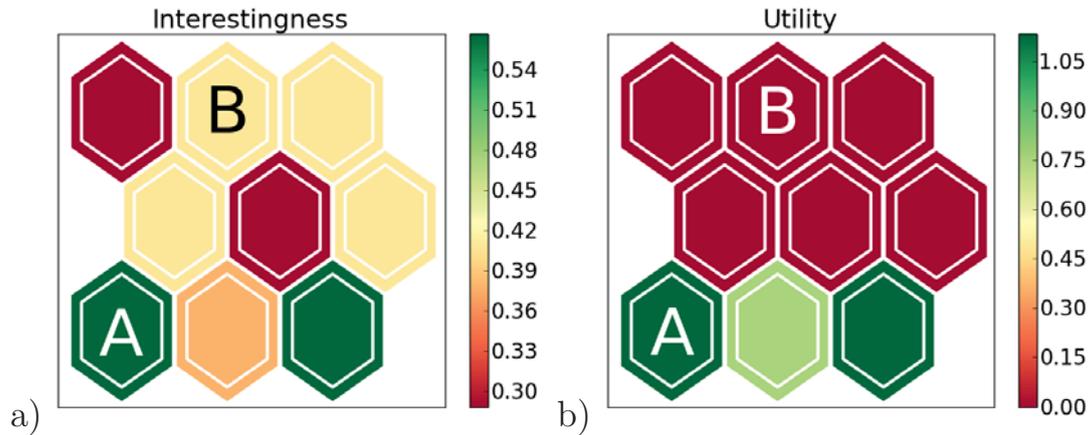
Although the SOM provides the possibility to visualize the structure of the high dimensional input data on a low dimensional map, an retrieval of abstract concepts from investigating the feature maps solely remains difficult, see [Ultsch, 1993]. Therefore the *Unified Distance Matrix* (U-Matrix) can be applied to further examine the organization of the feature vectors in the input space. In its simplest form the size of the U-Matrix equals the number of output units and provides information about the average distance of each unit to its neighbors. The extended U-Matrix additionally provides information about the individual distances between neighboring feature vectors and thus has almost twice the size of the SOM feature map.

Since the U-Matrix denotes similarities between features, it can be utilized to further group feature vectors into larger clusters. Sets of features with a high distance to all neighboring units form boundaries on the U-Matrix, separating the clusters. An automatic identification of associate features can



**Figure 5.5:** Results from the concept formation using self-organizing maps, where a) depicts the example dataset in the input parameter space, Figs. b) to d) visualize the projection of the input and output parameter to the output map of the SOM, e) depicts the location of the prototype vectors of the SOM in the input space and f) visualizes the distance between the prototype vectors in form of the unified distance matrix (U-Matrix). Prototypes A and B representing two distinct concepts, selected for a detailed analysis.

be performed by identifying the boundaries on the U-Matrix or by explicitly clustering the features based on their related distances. In [Pateritsas et al., 2007], the authors summarize both approaches and describe the emerged clusters by IF-THEN rules. A rule based abstraction from self-organizing maps can only approximate the emerged concept boundaries. While SOMs



**Figure 5.6:** Visualization of the IS a) and utility b) map. The utility map denotes the most relevant concepts, where one concept is represented by one neuron on the feature map.

represent concept boundaries as voronoi regions, the considered rule formulations represent boundaries as hyper-cubes in the parameter space. However, the association rules can directly be interpreted and processed by engineers and expert systems.

Fig. 5.4 summarizes the process for concept retrieval, rule extraction and concept evaluation based on self-organizing maps. Before the training of the SOM, a normalization of the meta design training data might be necessary. Subsequent to the training, optionally, high level concepts can be formed by means of further clustering the SOM prototype vectors. The final concepts are the basis for rule extraction and concept evaluation.

### Example

As an example, the retrieval process depicted in Fig. 5.4 has been applied to the same example dataset which has been already used for the tree induction in Section 5.4.1. The training of the SOM results in a two dimensional feature map with a pre-defined size of  $3 \times 3$  output nodes. In order to mimic a supervised training, the target attribute  $Y$  has been used as an additional input parameter. Figs. 5.5 b) to d) visualize the respective feature values of the SOM prototype vectors on the feature map. Additionally, the location of the prototype vectors for each output neuron in the input space is depicted in Fig. 5.5 e). The relationship between  $X_1$ ,  $X_2$  and  $Y$  can be investigated by a visual inspection of the different maps. For example, the maps show that for high values of  $X_1$  and low values of  $X_2$  the target variable  $Y$  is expected to be  $Y = -1.0$ .

The U-Matrix in Fig. 5.5 depicts the distances between the prototype vectors in the input space. The distance information can be used to highlight which of the output neurons could be grouped together to form new high level concepts. However, in this example each output neuron is considered as one individual concept without considering the subsequent extraction of high level concepts. Thus, rules are extracted for each output neuron based on a simple strategy. The attributes  $X_1$  and  $X_2$  define the antecedent and the attribute  $Y$  the consequence of each rule. The respective intervals for each attribute are defined by the minimum and maximum feature value of the data samples that are covered by each output neuron. Exemplary, the rules extracted for concept  $A$  and  $B$  (See Figs. 5.5 b) to f)) are listed:

$$\begin{aligned} A : & \quad X_1 \in [-0.10, 0.10] \wedge X_2 \in [-0.40, -0.15] \rightarrow Y = -1, \\ B : & \quad X_1 \in [-0.40, -0.20] \wedge X_2 \in [0.20, 0.40] \rightarrow Y = 1. \end{aligned}$$

Based on the formulated rules, the  $IS$  measure and the measure of utility have been evaluated for each concept. In order to visualize the evaluation results, the relevance values are mapped onto the SOM feature map resulting in the  $IS$  and utility map, see Figs. 5.6 a) and b). The utility map facilitates a visual inspection and identification of the most relevant concepts. In this example, concept  $A$  has been identified as one of two concepts with highest degree of utility. The utility value for concept  $B$  is small and therefore identifies a concept which is less relevant concerning the pre-defined objective.

## 5.5 Utilizing Information about Design Concepts

---

The extracted concepts and formulated rules can be directly utilized within a knowledge based engineering system, see [Rocca, 2012] for an introduction, or to build up expert systems for distinct design problems. In combination with the universal design representation concepts linked to the holistic design can be formulated. As such, the acquired design concepts and their formulation in linguistic form can directly help engineers in decision making, beyond individual processes. Depending on the overall strategy, sparsely sampled areas in the design space can be further explored, or the information about outperforming concepts can be exploited to guide the design process. The analysis can lead to the discovery of new design concepts and hypothesis which can be validated in subsequent experimental or simulation studies. Furthermore, the clear description of such concepts can reveal relevant interrelations between design parts, domains and engineers, upon which communication strategies can be revisited. In computational optimization, global search algorithms

like evolutionary strategies typically employ strategy parameters to guide the search process. The a priori adaptation of the strategy parameters has a positive effect on the performance of the search algorithm as shown in [Kern et al., 2004]. The initialization of those strategy parameters prior to the optimization run, or the definition of optimization constraints based on the acquired knowledge could further increase its efficiency.

## 5.6 Application to Passenger Car Design

---

In the following experimental studies, design concepts are extracted by applying two machine learning techniques, the tree learner and the self-organizing map to the reduced features of passenger car dataset 4, see section 2.4. With that we like to emphasize the possibilities for an automatic extraction of car concepts based on the surface mesh design representation. With the application of the concept learning to the design data, shapes are grouped together which are similar with respect to their surface representation and their aerodynamic performance, abstracting new passenger car classes. For each concept crisp rules are extracted to describe the concepts in human readable form. It will be shown how the utility measure can be used to order the design concepts according to their relevance.

The used training dataset contains the design data from two optimization runs, where two representations  $\mathcal{R}_A$  and  $\mathcal{R}_B$  have been used. The design variations are represented by the reduced feature set  $A_0$  to  $A_5$ , see Section 4.4.4. The performance of the designs is defined by the drag and rear lift differences to the baseline car shape, namely  $\phi_{FD}$  and  $\phi_{FLR}$ . In the design and analysis process the objective is the minimization of both performance indicators.

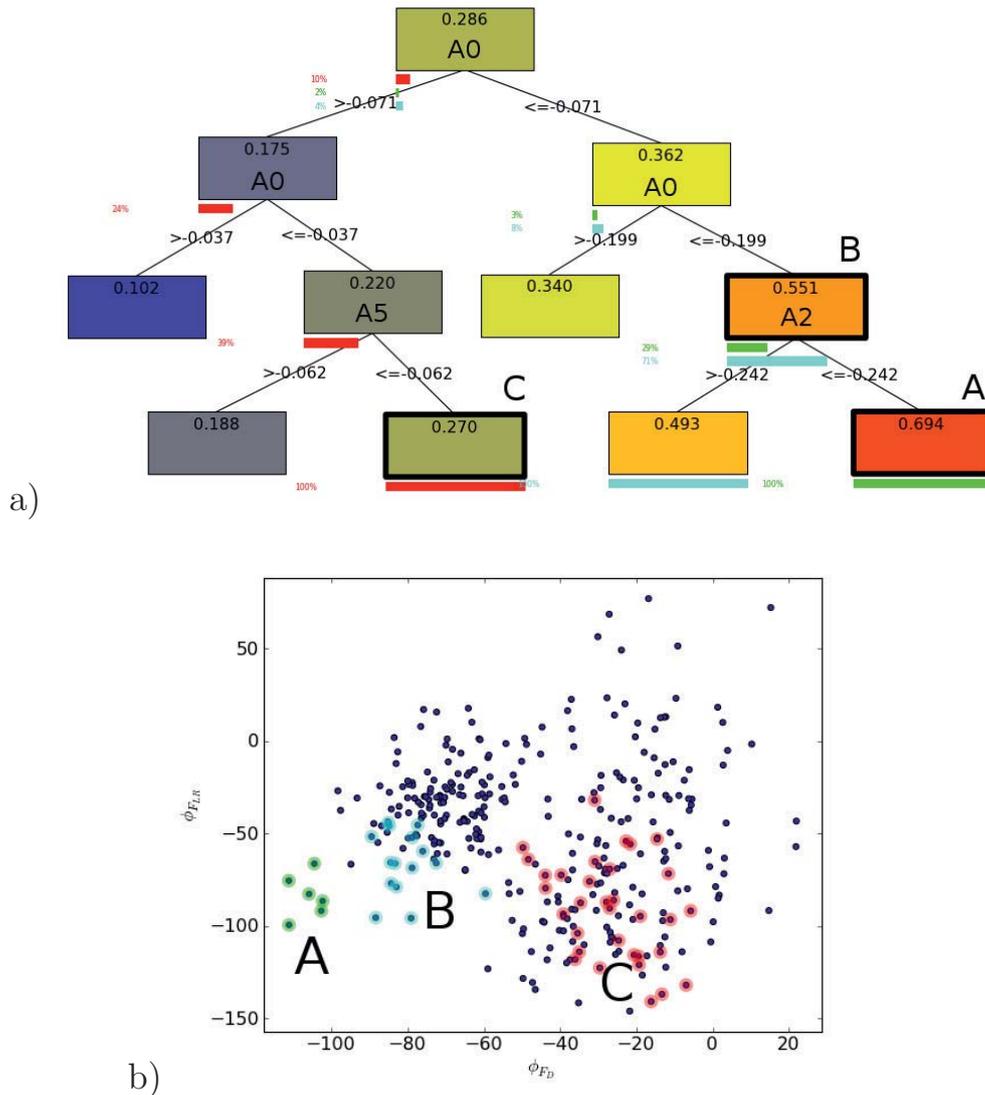
### 5.6.1 Tree Induction for Car Concept Retrieval

The applied tree learner is an implementation of the standard C4.5 tree induction algorithm. The gain ratio has been used to split the data samples at each node and thus grow the tree. If the gain ratio is lower than 0.2 or the number of samples represented by one node is below three the growth has been stopped. In addition, each node is restricted to binary splits. The independent parameters for the induction algorithm are the displacement values that correspond to the reduced cluster center vertices. The gain ratio for the split of the nodes is calculated based on a single scalar value. Therefore, the aerodynamic drag and the rear lift force have been combined into one characteristic value  $\phi_F$ , which is defined as the product between the normalized force differences:

## CHAPTER 5. DESIGN CONCEPT RETRIEVAL

$$\phi_F = \hat{\phi}_{FLR} \cdot \hat{\phi}_{FD}, \quad (5.9)$$

where  $\hat{\phi}_{FLR}$  and  $\hat{\phi}_{FD}$  define the respective normalized drag and rear lift force difference. The difference value  $\phi_F$  has been calculated for each design prior to the tree induction. The underlying training samples are depicted in the Fig. 5.7 b) with respect to their quality values.



**Figure 5.7:** a): Illustration of the result from the concept formation using the C4.5 tree induction algorithm. Three concepts A, B and C are marked and their corresponding designs are visualized in the objective space in plot b).

The resulting tree is illustrated in Fig. 5.7 a), where each node represents one potential car concept. The label at the top of each node captures the

## 5.6. APPLICATION TO PASSENGER CAR DESIGN

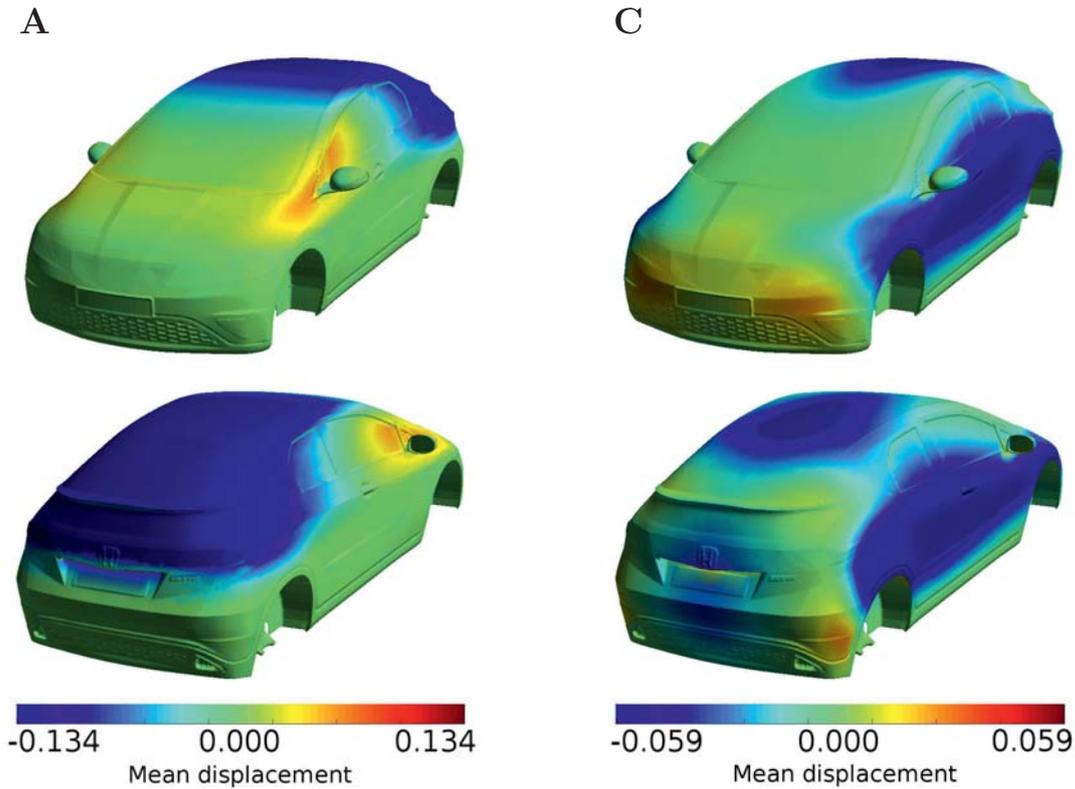
average value of  $\phi_F$  calculated based on the covered data samples. A background color is assigned to each node proportional to the average value of  $\phi_F$ . In addition, the internal nodes are labeled with the split variables relating to the features  $A0$  to  $A5$ . The split variable  $A0$ , which is representing shape variations around the roof, defines the variable which best separates the entire dataset with respect to  $\phi_F$ . It maximizes the gain ratio for the two resulting data subsets. The child node at the left and right hand side cover 40% and 60% of the entire data set, respectively. Both child nodes are again split based on the attribute  $A0$ . The data samples covered by each right child node are partitioned based on  $A5$  or  $A2$ , where  $A5$  is representing shape variations at the side and  $A2$  variations at the rear window of the car.

Three nodes  $A$ ,  $B$  and  $C$  have been selected from the tree, representing three different design concepts. The designs covered by each concept are highlighted in color in Fig. 5.7 b). The colored bars below selected nodes in the tree show the respective distribution of the samples in the parent nodes. For each of the three concepts rules are extracted from the tree to represent the design concepts in linguistic form. The rules are as follows:

$$\begin{aligned}
 A & : A0 \in [-0.29, -0.20] \wedge A2 \in [-0.27, -0.24] \rightarrow \\
 & \quad \phi_{FD} \in [-111.25, -102.40] \wedge \phi_{FLR} \in [-99.50, -66.31], \\
 B & : A0 \in [-0.29, -0.20] \rightarrow \\
 & \quad \phi_{FD} \in [-111.25, -59.65] \wedge \phi_{FLR} \in [-99.50, -44.10], \\
 C & : A0 \in [-0.07, -0.04] \wedge A5 \in [-0.11, -0.06] \rightarrow \\
 & \quad \phi_{FD} \in [-49.83, -5.81] \wedge \phi_{FLR} \in [-140.92, -31.91].
 \end{aligned}$$

The rules describe general associations between surface deformations with respect to the reference design and the expected objective values. As an example, rule  $B$  describes that a deformation of the surface patch around the roof  $A0$  to the inside of the surface between 29 and 30 cm will lead to a respective drag reduction of  $-111.25$  N to  $-59.65$  N and a rear lift reduction of  $-99.50$  N to  $-44.10$  N.

In order to analyze the shape differences between different concepts the mean displacements to the baseline surface for concept  $A$  and  $C$  have been calculated. The mean displacements are color coded and projected onto the surface of the baseline design. Fig. 5.8 shows the results for a frontal and rear view to the car shape. The range of the shown mean displacements is denoted in the respective color bars. It can be seen that concept  $A$  covers mostly large deformations around the rear of the car shape while concept  $C$  covers deformations at the roof and the side of the car.



**Figure 5.8:** Comparison of the mean surface displacement of concept *A* and *C*, calculated with respect to the passenger car baseline design.

	#designs	IS	vol	util
A:	6	1.00	0.79	<b>0.79</b>
B:	21	0.65	0.79	<b>0.51</b>
C:	33	0.52	0.48	<b>0.25</b>

**Table 5.2:** Detailed results from the concept evaluation for concepts *A*, *B* and *C*, derived from the decision tree.

Due to the constraints on the growth of the tree the number of nodes and branches has been kept small. Relaxing the constraints can result in larger trees and as consequence in more complex rules. Especially for larger or multiple trees that are generated from different datasets, an analysis of all nodes can get laborious and the ranking or filtering of concepts is desired. To emphasize the benefit of the introduced utility measure, concepts *A* to *C* have been evaluated. The results are summarized in data Tab. 5.2.

The concept evaluation assigns the highest utility value to concept *A* followed by *B* and *C*. The hyper-volume of concept *A* and *B* is the same, since both concepts share the non-dominated design solutions. However, the *IS* value by means of the correlation between design variations and changes

in the objective values is lower for concept  $B$  and thus, concept  $B$  is of lower utility compared to  $A$ .

### 5.6.2 Car Concept Learning with SOMs

As alternative to the tree learning, an partitioning of the design dataset has been carried out using the SOM algorithm. The input parameters for the artificial neural network are again the reduced set of shape features  $A0$  to  $A5$ . Using the same dataset as in the tree induction process, the design data of dataset 4 has been used as training data for the SOM algorithm. In order to mimic a supervised clustering with the SOM algorithm, the performance number  $\phi_F$  (see equation (5.9)) has been added as an additional input. The structure of the feature map has been pre-defined.  $5 \times 5$  output neurons are regularly arranged on a 2d feature map. The weight vectors of the  $5 \times 5$  feature map are adapted based on the Euclidean distance between the input samples. After learning, each weight vector represents one prototype vector for one potential design concept.

The resulting topographic maps are shown in Fig. 5.9. The individual patches are colored with the weight values for different shape features  $A0$  to  $A5$  and  $\phi_F$  respectively. For example, a strong correlation between the maps of  $A1$  and  $A2$ , and a strong negative correlation between  $A0$  and the objective value  $\phi_F$  can be observed. In addition the U-Matrix has been calculated and is depicted in Fig. 5.9. The U-Matrix shows the distance between the weight vectors in the input space and can provide information to further group individual concepts into high level concepts. Thus, e.g., the first two lines of output neurons have a large distance to the remaining ones and could be further grouped together. However, we consider each weight vector as a prototype for a potential concept. Three concepts  $D$ ,  $E$  and  $F$  have been selected and the corresponding samples are visualized in the objective space in Fig. 5.10. The designs which are covered by each of the three concepts are highlighted in color.

The detailed associations of the three selected concepts between design parameters and the differences in the objective values  $\phi_{F_D}$  and  $\phi_{F_{LR}}$  are as follows:

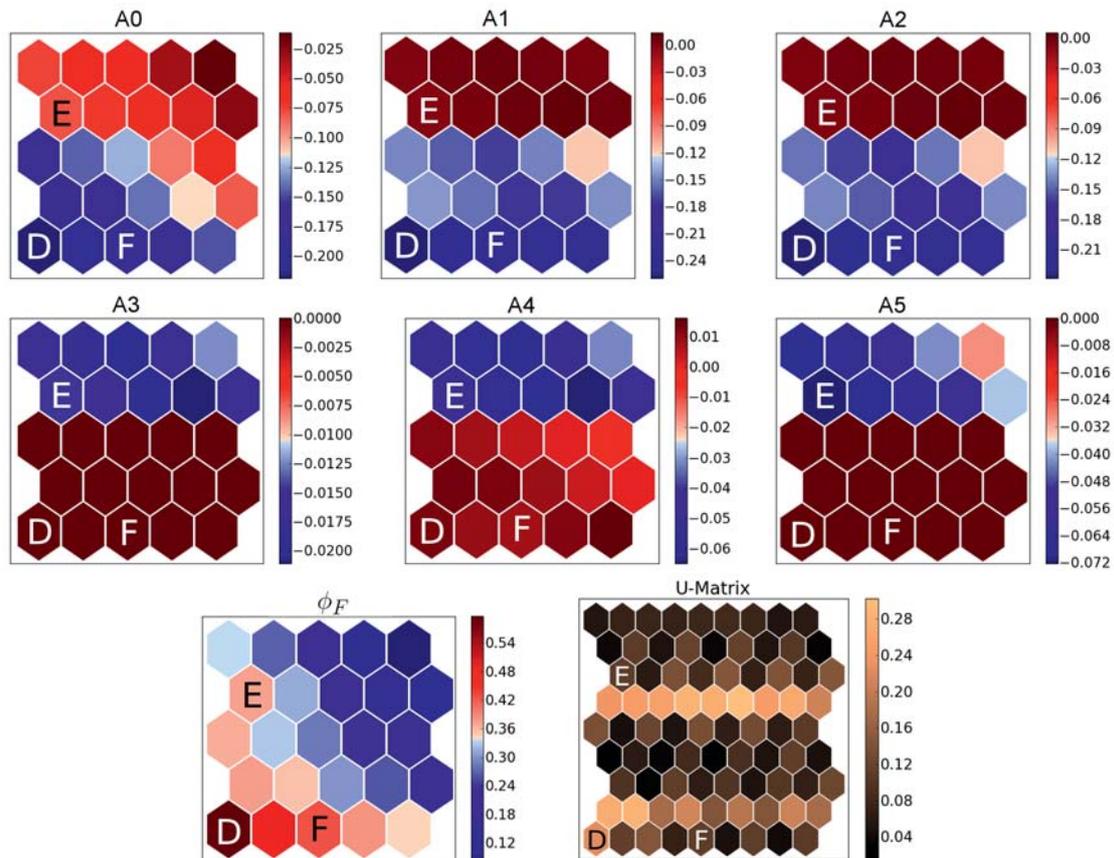
$$\begin{aligned}
 D & : A0 \in [-0.29, -0.18] \wedge A1 \in [-0.31, -0.22] \wedge \\
 & \quad A2 \in [-0.27, -0.20] \wedge A3 \in [0.00, 0.00] \wedge \\
 & \quad A4 \in [0.00, 0.03] \wedge A5 \in [0.00, 0.00] \rightarrow \\
 & \quad \phi_{FD} \in [-111.25, -79.17] \wedge \phi_{FLR} \in [-99.50, -66.31], \\
 E & : A0 \in [-0.11, -0.06] \wedge A1 \in [-0.05, 0.03] \wedge \\
 & \quad A2 \in [-0.05, 0.01] \wedge A3 \in [-0.02, 0.00] \wedge \\
 & \quad A4 \in [-0.07, 0.00] \wedge A5 \in [-0.09, -0.05] \rightarrow \\
 & \quad \phi_{FD} \in [-59.02, -35.05] \wedge \phi_{FLR} \in [-141.63, -82.59], \\
 F & : A0 \in [-0.22, -0.15] \wedge A1 \in [-0.26, -0.17] \wedge \\
 & \quad A2 \in [-0.24, -0.16] \wedge A3 \in [0.00, 0.00] \wedge \\
 & \quad A4 \in [-0.01, 0.03] \wedge A5 \in [0.00, 0.00] \rightarrow \\
 & \quad \phi_{FD} \in [-98.38, -64.97] \wedge \phi_{FLR} \in [-69.58, -26.89],
 \end{aligned}$$

The parameter range of each design parameter is used to formulate the antecedent part of the association. The range for each feature is defined by the minimum and maximum of the covered feature values. Typically, parameters with vanishing variances can be removed from the association, requiring the implementation of an additional post-processing step. For example, the antecedent of rule  $D$  contains three features  $A3$ ,  $A4$  and  $A5$  which could be neglected. The defined range can describe only a minimum number of designs or even none. Thus, compressing the information, the antecedent of rule  $D$  describes those designs which have a certain deformation at the rear or the roof of the car. With that, concept  $D$  is comparable with concept  $A$  from Section 5.6.1, but covers 11 instead of 6 designs.

For comparison and a more detailed analysis of the selected concepts the mean displacements to the baseline surface have been calculated for concept  $D$  and  $E$ . The frontal and rear views onto the baseline car with projected mean displacement values are shown in Fig. 5.11. Again, the range of the shown mean displacements is denoted in the respective color bars. Analyzing the mean displacement can reveal detailed differences between the concepts. It can be seen that concept  $D$  and  $A$  as well as concept  $E$  and  $C$  are comparable (see Fig. 5.8 for reference). However, the comparison also reveal some local differences between the concepts.

All in all, the SOM represents 25 concepts. In order to rank the individual concepts the utilities have been calculated for each concept and visualized in the *utility map*, see Fig. 5.12 a). Without analyzing the topographic maps of all objective values, the utility map provides a quick visual summary of

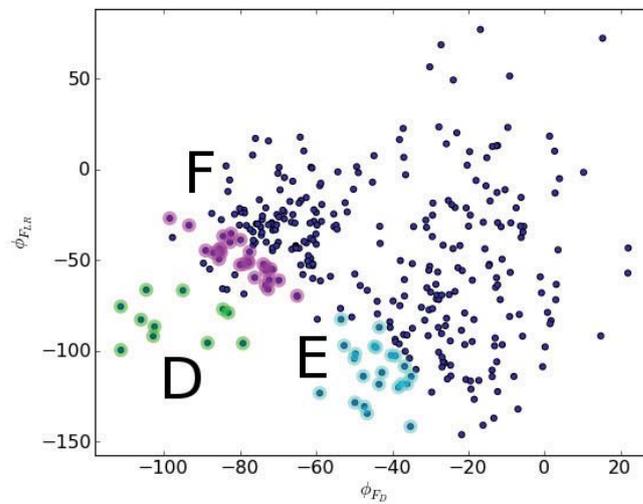
## 5.6. APPLICATION TO PASSENGER CAR DESIGN



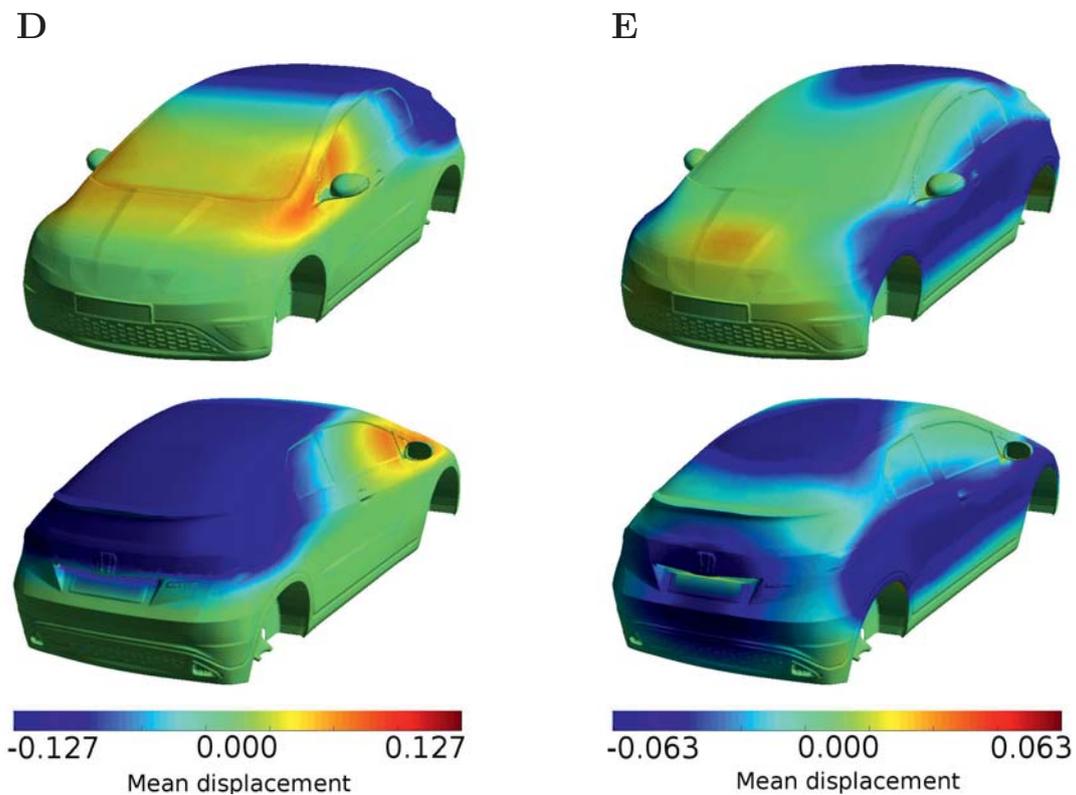
**Figure 5.9:** Illustration of the results from concept formation using the SOM algorithm. The weight values for the different inputs A0 to A5 and  $\phi_F$  are shown, together with the U-Matrix. Exemplary, three concepts D, E and F are labeled in each plot for which linguistic rules are formulated.

the most relevant concepts. The three concepts A, B and C are the ones with highest degree of utility and they are silhouetted from the remaining 22 concepts on the utility map. For more detailed insights, the *IS*, the hyper volume, the confidence and the coverage within each concept are shown in Figs. 5.9 b) to e), e.g., depicting that concept B has a higher correlation between design parameters and objective values, but a lower hyper volume compared to concept A.

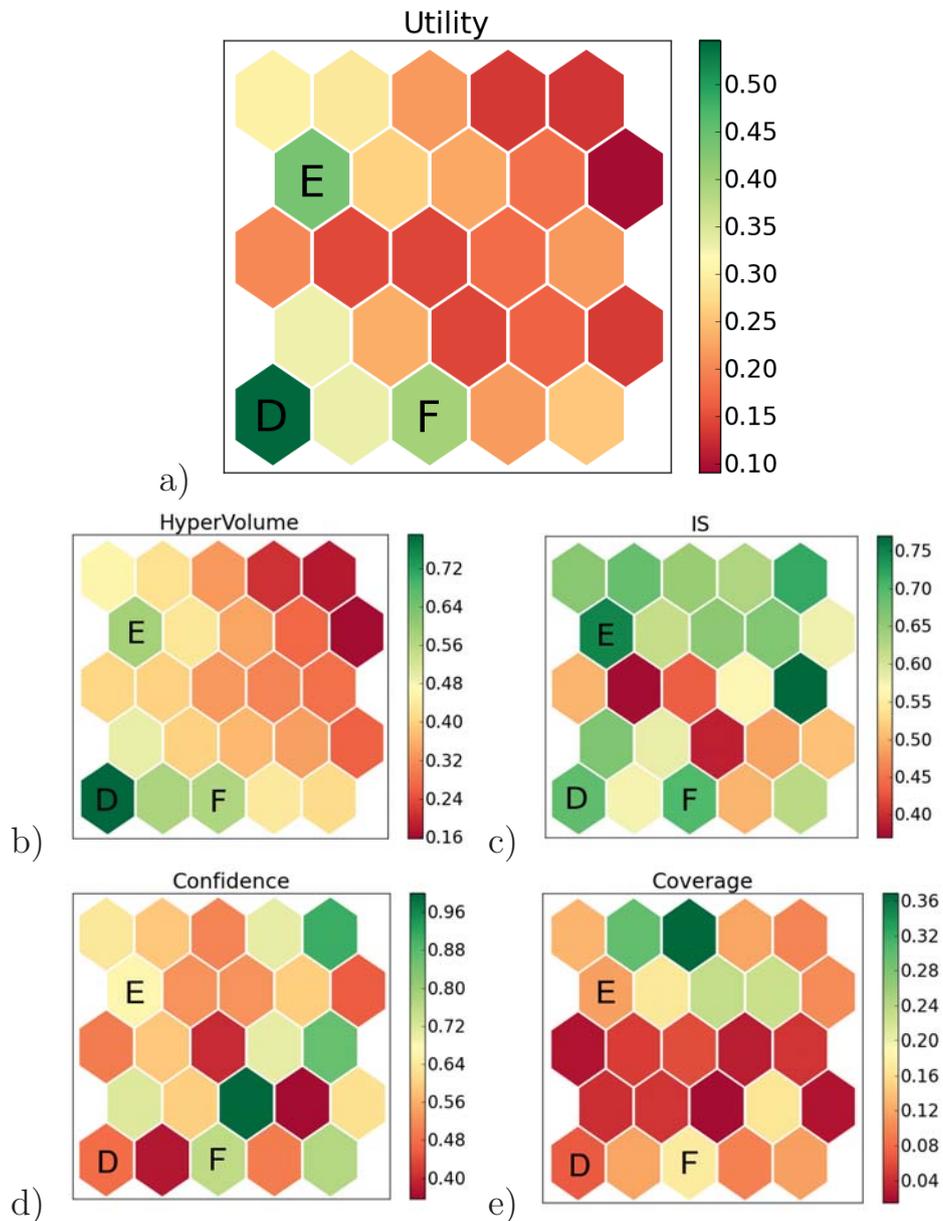
Either using the tree induction, SOMs or alternative techniques from machine learning and data mining engineers can extract new concepts explaining relations in the design domain. By adding the concepts to a global knowledge base the knowledge can be shared among engineers and utilized to improve future designs. Wherein, the utility measure can help to sort concepts according to each engineer's objectives, as shown in our experimental studies.



**Figure 5.10:** Visualization of the training data in the objective space. Three concepts *D*, *E* and *F* are labeled, highlighting the designs covered by each of the three concepts.



**Figure 5.11:** Comparison of the mean displacements for concept *D* (left) and concept *E* (right), calculated with respect to the passenger car baseline shape.



**Figure 5.12:** Visualization of the utility map a) depicted to allow the identification of the most relevant concepts. For completeness the hypervolume b), the IS value c), the confidence d) and the coverage e) for each concept are shown.



### 5.7 Interim Summary

---

In this chapter the procedure for the retrieval, description and evaluation of design concepts has been generalized and can be carried out independently of the used modeling technique. Exemplary two modeling techniques, the decision tree algorithms and the self-organizing map have been studied in more detail. A new measure has been introduced to evaluate extracted design concepts based on the estimation of their utility. The new measure allows the ranking of concepts according to the formulation of the engineer's objectives.

Given the reduced set of shape features, introduced in Chapter 4, the automatic concept retrieval process has been applied to the passenger car design data. The results denote that the automatic abstraction of design concepts from unstructured surface meshes is feasible. Thus, it provides a powerful tool for the holistic analysis of design data in an integrative or corporate design process. Augmenting the new utility measure allows the formulation of a reduced set of linguistic rules which ease the transfer of knowledge to engineers and expert systems. Furthermore, the algorithmic extraction of design concepts allows the wider distribution of engineering knowledge in a company.



# 6

## Interaction Analysis

When the complexity of a design and the corresponding design process increases, a decomposition into sub-components is often the only possibility to efficiently manage the design and optimization process. *Interaction Analysis* targets the computer-aided identification and classification of interrelations among sub-components of a design. In this chapter the *interaction information* an information theoretic attempt for the quantification of interactions among design parts are studied. So called *interaction graphs* are adopted to visualize the results of the interaction analysis to the engineers. Based on artificial examples, properties of the three-way interaction information are investigated, which aid a proper interpretation of the results from the interaction analysis of the passenger car design data.

### 6.1 Design Interaction

---

In order to handle complex design problems, the design is often composed of smaller sub-components which are designed and optimized separately and in parallel. The interplay between the sub-components plays a crucial role for the behavior and performance of the overall design. As an example, the influence of a Formula One car's rear wing on the overall downforce of the car depends strongly on the correct adjustment of the frontal part of the car, especially with respect to the front wing angle. The identification and analysis of such interactions is an important task to find an effective system decomposition, which allows to improve the overall system behavior by an optimization of the individual sub-components. In practice, the decomposition into sub-components is mostly done based on engineers' experiences, it remains fixed over the course of the design process, and it often does consider only structural design properties. A proper problem decomposition strategy should rather take interactions between the sub-components together with their influence on the design performance into account, and



## CHAPTER 6. INTERACTION ANALYSIS

---

should furthermore consider to restructure the overall system with changes of the system requirements or its working conditions. If interactions between sub-components are not properly considered, the final fusion of the sub-components can lead to a significant drop in the overall design efficiency and can hinder the success of the entire design process.

The interaction analysis targets an automatic identification and classification of interrelations between sub-components, allowing to restructure or verify the overall system decomposition. For this purpose, the interaction analysis investigates the interplay between different components based on a given design representation and performance evaluation, where the design representation and the corresponding parameters define the structure and the performance characterizes the behavior of a system. Ashby's and Krippendorff's definition of interaction, published in [Krippendorff, 1986, p. 36 et seq.], has been adopted and extended to a working definition of a *Design Interaction*:

**Design Interaction** Design interaction is defined as the unique dependency between multiple design parameters with respect to the overall design performance, from which all dependencies related to a design parameter subset are removed, wherein design parameters and the design performance denote specific attributes of a system.

Based on a specific design dataset, the result of the interaction analysis should provide information about which parameters and parameter sub-sets are correlated or redundant with respect to a pre-defined performance criteria. In contrast to the sensitivity analysis of Chapter 4, the interaction analysis does not target to quantify the overall influence of individual parameters, rather it quantifies the influence of joint parameters on the design performance as a unique property of the considered design parameter set.

In statistics, existing algorithms for the identification of interactions mostly relate to methods for regression analysis and variance decomposition, see [Jaccard and Turrisi, 2003] and [Saltelli et al., 2007], respectively. In terms of regression analysis, as stated in [Jaccard and Turrisi, 2003], "An interaction effect is said to exist when the effect of the independent variable on the dependent variable differs depending on the value of a third variable, called the moderator variable.", wherein the independent variables are representing observations of the cause (e.g design parameters) for the change of one or multiple dependent variables (e.g., performance numbers). To model such interactions, so called interaction terms have been introduced. They model the respective moderated causal relationships between variables. The coefficients of the interaction terms, estimated using the least squares computation, together with their statistical significance provide information about



## 6.2. FUNDAMENTAL INFORMATION CONCEPTS

---

the relevance of modeled interactions. For the setup of the regression model, the functional relation between the moderator variable and its influence on the moderated variable needs to be pre-defined, which strongly limits the application of regression models for the identification of arbitrary interactions of unknown characteristic.

In contrast to regression analysis, methods of variance decomposition, see [Saltelli et al., 2007], provide a "model-free" approach to measure the sensitivity of parameter interactions. Those methods estimate the amount of variance of a certain dependent variable that is associated to variations of the independent variables. The overall variance of the output variable is decomposed into variance contributions from the main and interaction effects of the considered input variables. As such the relevance of interaction effects is determined by the proportion of the related variance that add to the overall variance of the output variable. Although, methods of variance decomposition are an effective approach to identify linear as well as non-linear interactions, their main drawback is the requirement for statistical independence among the input variables. Especially, for the analysis of data from various design processes, this can hardly be guaranteed.

In the remainder of this chapter the so called *interaction information* is studied and applied to analyze design interactions. Derived from Shannon's theory of information, the interaction information models the interrelation between attributes in terms of information transmission between communication channels. In contrast to the regression analysis with interaction terms and the interaction analysis by variance decomposition, the information theoretic attempt per se makes no assumptions on the distribution or the interrelation between variables, and can equally be applied to quantitative and qualitative data.

## 6.2 Fundamental Information Concepts

---

The following list provides a concise overview of the most relevant information concepts, which are the basis for the definition and interpretation of the interaction information. A more detailed introduction into information theory and the related terminology can be found in [Abramson, 1963] and [MacKay, 2003, Chapt. 4&8]. The Shannon Entropy and the mutual information have already been introduced in Section 4.1 of this thesis. While Section 4.1 focused on the study of robust aspects of the information measures, this section serves as basis for the calculation of multi-variate information measures as a pre-requisite for a probabilistic analysis of parameter interactions.



- **Shannon Entropy:**

The Shannon entropy quantifies the uncertainty about the prediction of a discrete random variable  $X$ , and is defined based on the probability masses  $p(x)$  as:

$$H(X) = - \sum_{x \in \mathbb{X}} p(x) \log_2 p(x),$$

with  $0 \leq H(X) \leq \log_2 |\mathbb{X}|$ , where  $|\mathbb{X}|$  denotes the number of discrete values  $X$  can take.

- **Joint Entropy:**

The joint entropy defines the Shannon entropy for multiple variables based on their related joint probability distribution. Given a set of  $n$  discrete random variables,  $\mathcal{X} = \{X_1, \dots, X_n\}$ , the joint entropy is defined as:

$$H(\mathcal{X}) = - \sum_{\substack{x_1 \in \mathbb{X}_1 \\ \vdots \\ x_n \in \mathbb{X}_n}} p(x_1, \dots, x_n) \log_2 p(x_1, \dots, x_n),$$

with  $0 \leq H(\mathcal{X}) \leq \log_2 (|\mathbb{X}_1| \cdot |\mathbb{X}_2| \cdot \dots \cdot |\mathbb{X}_n|)$ .

- **Conditional Entropy:**

Simultaneously considering two sets of random variables,  $\mathcal{X} = \{X_1, \dots, X_n\}$  and  $\mathcal{Y} = \{Y_1, \dots, Y_m\}$ , the conditional entropy estimates the remaining expected uncertainty of a set of random variables  $\mathcal{Y}$ , if the probabilities of a second set of random variables  $\mathcal{X}$  are already known:

$$\begin{aligned} H(\mathcal{Y}|\mathcal{X}) &= \sum_{\substack{x_1 \in \mathbb{X}_1 \\ \vdots \\ x_n \in \mathbb{X}_n}} p(x_1, \dots, x_n) H(\mathcal{Y}|\mathcal{X} = \{x_1, \dots, x_n\}) \\ &= H(\mathcal{X} \cup \mathcal{Y}) - H(\mathcal{X}), \end{aligned} \tag{6.1}$$

with

$$\begin{aligned} H(\mathcal{Y}|\mathcal{X} = \{x_1, \dots, x_n\}) &= \\ &= - \sum_{\substack{y_1 \in \mathbb{Y}_1 \\ \vdots \\ y_m \in \mathbb{Y}_m}} p(y_1, \dots, y_m | x_1, \dots, x_n) \log_2 p(y_1, \dots, y_m | x_1, \dots, x_n). \end{aligned}$$

The conditional probability  $p(y_1, \dots, y_m | x_1, \dots, x_n)$  can be calculated in terms of joint probabilities as:

$$p(y_1, \dots, y_m | x_1, \dots, x_n) = \frac{p(y_1, \dots, y_m, x_1, \dots, x_n)}{p(x_1, \dots, x_n)},$$

with

$$p(x_1, \dots, x_n) = \sum_{\substack{y_1 \in \mathbb{Y}_1 \\ \vdots \\ y_m \in \mathbb{Y}_m}} p(x_1, \dots, x_n, y_1, \dots, y_m).$$

The conditional entropy  $H(\mathcal{Y}|\mathcal{X}) = 0$  if and only if  $\mathcal{Y}$  is completely determined by  $\mathcal{X}$ . If all variables of  $\mathcal{Y}$  and  $\mathcal{X}$  are independent, then  $H(\mathcal{Y}|\mathcal{X}) = H(\mathcal{Y})$ .

- **Mutual Information:**

Considering two sets of random variables,  $\mathcal{X} = \{X_1, \dots, X_n\}$  and  $\mathcal{Y} = \{Y_1, \dots, Y_m\}$ , the mutual information quantifies the information shared between  $\mathcal{X}$  and  $\mathcal{Y}$ . Therefore, it compares the approximated joint entropy under the assumption of statistical independence ( $H(\mathcal{X}) + H(\mathcal{Y})$ ) with the observed joint entropy  $H(\mathcal{X} \cup \mathcal{Y})$ . As such, the mutual information is defined as the difference between approximated and observed joint entropy:

$$I(\mathcal{X}; \mathcal{Y}) = H(\mathcal{X}) + H(\mathcal{Y}) - H(\mathcal{X} \cup \mathcal{Y}), \quad (6.2)$$

with  $0 \leq I(\mathcal{X}; \mathcal{Y}) \leq \max\{H(\mathcal{X}), H(\mathcal{Y})\}$ .  $I(\mathcal{X}; \mathcal{Y}) = 0$  only if the variables of  $\mathcal{X}$  and  $\mathcal{Y}$  are statistically independent, so that  $H(\mathcal{X}) + H(\mathcal{Y}) = H(\mathcal{X} \cup \mathcal{Y})$ .

- **Conditional Mutual Information:**

Given three sets of random variables,  $\mathcal{X} = \{X_1, \dots, X_n\}$ ,  $\mathcal{Y} = \{Y_1, \dots, Y_m\}$  and  $\mathcal{Z} = \{Z_1, \dots, Z_n\}$ , the conditional mutual information is a measure of the information shared between two sets  $\mathcal{X}$  and  $\mathcal{Y}$ , if the variables of a third set  $\mathcal{Z}$  are already known. In terms of joint entropies, the conditional mutual information can be defined as:

$$I(\mathcal{X}; \mathcal{Y} | \mathcal{Z}) = H(\mathcal{X} \cup \mathcal{Z}) + H(\mathcal{Y} \cup \mathcal{Z}) - H(\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}) - H(\mathcal{Z}), \quad (6.3)$$

with  $I(\mathcal{X}; \mathcal{Y} | \mathcal{Z}) \geq 0$ , where  $I(\mathcal{X}; \mathcal{Y} | \mathcal{Z})$  can be larger or smaller than  $I(\mathcal{X}; \mathcal{Y})$ .

The mutual information and the conditional mutual information can be reformulated in terms of conditional entropies by substituting the joint entry in equation (6.2) and (6.3) with equation 6.1.

### 6.3 Interaction Information

---

The mathematical formulation of the interaction information provides a general probabilistic attempt for the quantification of interaction effects for multiple attributes. For this purpose the interaction information investigates the co-occurrence of discrete values of the related random variables. The interaction information has first been introduced as an extension of the mutual information to three random variables by [McGill, 1954]. Later, Ashby and others extended his idea to more than three variables, see [Krippendorff, 1986, 2009a]. More recently, in [Jakulin and Bratko, 2003, 2004a], the authors reconsidered the formulation of the interaction information to identify the most significant parameter and parameter interactions, in order to increase the efficiency of classification models. The author generalized the interaction information to a set of multiple random variables, referred to as  $\mathcal{X} = \{X_1, \dots, X_n\}$ , and presented a closed mathematical formulation of the interaction information based on an iterating sum over marginal and joint entropies:

$$Q(\mathcal{X}) = - \sum_{\mathcal{T} \subseteq \mathcal{X}} (-1)^{|\mathcal{X}|-|\mathcal{T}|} H(\mathcal{T}). \quad (6.4)$$

where  $\mathcal{T}$  defines each possible subset of the attributes contained in  $\mathcal{X}$ . For  $|\mathcal{X}| = n$  variables the interaction information is denoted as  $n$ -way interaction information. For two attributes, with  $\mathcal{X} = \{X_1, X_2\}$ ,  $Q(X_1, X_2)$  quantifies the so called two-way interaction:

$$Q(X_1; X_2) = I(X_1; X_2) = H(X_1) + H(X_2) - H(\{X_1, X_2\}),$$

which is identical to the mutual information for two variables, see equation (6.2).

However, it is important to note that in contrast to the mutual information, the interaction information can get negative values for more than two variables. The interpretation of negative interaction values has been discussed in the recent years in the related literature [Krippendorff, 2009a,b; Leydesdorff, 2009]. While the interaction information has first been introduced as a direct extension of the mutual information by [McGill, 1954], Krippendorff in [Krippendorff, 2009a] argued against the interaction information as a valid information theoretic quantity. He concluded that the interaction information can not be interpreted in the Shannon sense because it violates Shannon's definition of a strict relationship between variables (more precisely between sender and receiver in terms of communication theory). Therefore,

more recently the term  $Q$  instead of Jakulin's  $I$  has been used, see [Krippendorff, 2009a] and [Leydesdorff, 2009] for further information. "However, it does not conflict with a statistical approach that uses information theory for the measurement of specific associations in a non-linear dynamics", as summarized by [Leydesdorff, 2009]. Throughout the thesis I follow Leydesdorff, and treat the interaction information as a statistical approach for quantifying associations between random variables. In this work the interaction information is considered not to be a direct measure of Shannon's type of information, but rather a quantity that measures the information balance between multiple system variables. In practice, the estimation of higher order interactions can get difficult because the limited number of given data often hinders a reliable estimation of the high-dimensional joint probability distributions. Therefore, the focus of the subsequent section is on the formulation and interpretation of the three-way interaction information. The study of the three-way interaction information and its interpretation reveals the major concept and characteristics of the interaction information, which can equally be extended to higher order interactions within future studies.

## 6.4 Three-way Interaction

---

The interaction information for three random variables  $\mathcal{X} = \{X_1, X_2, X_3\}$ ,  $Q(X_1; X_2; X_3)$ , measures the information content resulting from a combination of the three variables, which is not present in any of the two-way interactions. According to equation (6.4), the three-way interaction in terms of marginal and joint entropies is defined as:

$$Q(X_1; X_2; X_3) = H(\{X_1, X_2\}) + H(\{X_2, X_3\}) + H(\{X_1, X_3\}) - H(X_1) - H(X_2) - H(X_3) - H(\{X_1, X_2, X_3\}) \quad (6.5)$$

Reformulating the interaction information in terms of information transfer eases the interpretation of  $Q(X_1; X_2; X_3)$ . Considering the definition of the mutual information from Section 6.2, equation (6.5) can, e.g., be reformulated in terms of information differences as:

$$Q(X_1; X_2; X_3) = I(\{X_1, X_2\}; X_3) - I(X_1; X_3) - I(X_2; X_3). \quad (6.6)$$

As such,  $Q(X_1; X_2; X_3)$  defines the expected amount of information that  $X_1$  and  $X_2$  convey about  $X_3$ , reduced by the information  $X_1$  and  $X_2$  individually share with  $X_3$ . Thus,  $Q(X_1; X_2; X_3)$  quantifies the amount of

information which is unique to the joint observation of all attributes. Since  $Q(X_1; X_2; X_3)$  is symmetric, this holds for any permutation of  $X_1$ ,  $X_2$  and  $X_3$ .

As an alternative formulation, substituting  $I(\{X_1, X_2\}; X_3)$ , e.g., with its mathematical equivalent  $I(X_1; X_3|X_2) + I(X_1; X_3)$ ,  $Q(X_1; X_2; X_3)$  can equally be written in terms of conditional mutual information as:

$$Q(X_1; X_2; X_3) = I(X_1; X_3|X_2) - I(X_1; X_3), \quad (6.7)$$

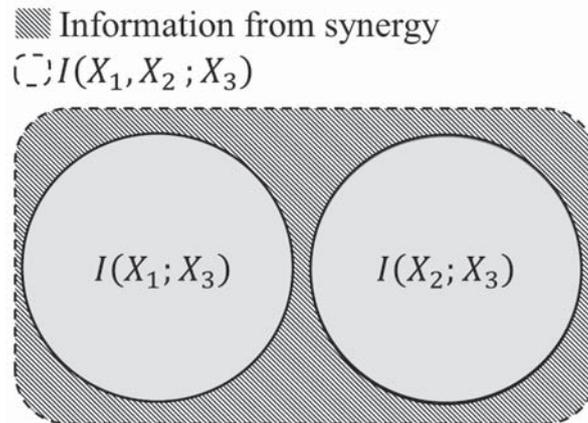
so that  $Q(X_1; X_2; X_3)$  is formulated as the difference between the information shared between  $X_1$  and  $X_3$ , given  $X_2$  is known or not.

Depending on the sign of the interaction value, Jakulin has classified  $Q(X_1; X_2; X_3)$  into three classes [Jakulin and Bratko, 2004b] and has interpreted them in the following way. If  $Q(X_1; X_2; X_3) < 0$ , *redundancy* is dominating the three-way interaction, e.g., the knowledge of  $X_2$  adds redundant information to the information shared between  $X_1$  and  $X_3$ . In contrast, if  $Q(X_1; X_2; X_3) > 0$  the interaction is characterized as *synergy*. In the case where  $Q(X_1; X_2; X_3) = 0$ , it is said that no three way interaction (*non-interaction*) among the variables is present. In the following, redundancy, synergy and non-interaction types are studied in more detail, in order to derive a better understanding of the three different classes and the related interaction values. Simple examples are given to reveal the circumstances under which an interaction can be assigned to either of the three classes. Furthermore, the *compensation* aspect is explained, which complicates a distinct assignment of the measured interaction values to one of the three classes.

### 6.4.1 Synergy

The interaction information  $Q(X_1; X_2; X_3)$  becomes positive if the joint information  $I(\{X_1, X_2\}; X_3)$  exceeds the sum of the information that each variable  $X_1$  and  $X_2$  conveys about the third variable  $X_3$ . This means that the observation and consideration of both random variables  $X_1$  and  $X_2$  simultaneously provides more information about  $X_3$  than considering each of the variables and its shared information with  $X_3$  separately. In this sense, information about  $X_3$  is gained from the synergy of the variables. Fig. 6.1 is sketching this circumstance, where the dashed area corresponds to the surplus of information gained from synergy. Equally, all these considerations hold for any permutation of  $X_1$ ,  $X_2$  and  $X_3$ .

Let's consider a task with the target to find a model, which best predicts  $X_3$  given  $X_1$  and  $X_2$ . The quality of the prediction can only be as good as the amount of information that is provided to the model through



**Figure 6.1:** *Diagram illustrating the interrelation between the mutual informations and the joint information, which result in a positive interaction information proportional to the dashed area.*

its structure. From the positive interaction information it can be inferred, that if  $X_3$  is modeled by two independent models, one with input  $X_1$  and the another one with input  $X_2$ , is worse compared to a single model that explicitly models the interrelation between  $X_1$  and  $X_2$ . As a consequence the attributes related to  $X_1$  and  $X_2$  should in general not be modeled as separate sub-components. In fact, this can be transferred to the design representation and related optimization processes. If a positive interaction is observed between sub-components of a system, the sub-components should be modeled as one system. Splitting them would make a correct prediction, e.g., of the design performance difficult, and thus can complicate the overall design and optimization process.

**Example** The following toy example should illustrate the circumstances under which the interaction information becomes positive. Lets consider a passenger car where the acceleration force  $F$ , the weight  $W$  and the engine power  $P$  are given as random variables. Furthermore, lets consider that the physical law, which is describing the interrelation between the individual variables is unknown. As depicted in Fig. 6.2, for the different variables qualitative values are observed, classifying  $W$ ,  $P$  and  $F$  into low ( $l$ ), medium ( $m$ ) and high ( $h$ ). From the calculation of the mutual information  $I(W; F)$  and  $I(P; F)$ , see Fig. 6.2, it is observed that weight and engine power alone, each explains about 33% of the uncertainty of  $F$ , with  $H(F) = 1.500$  bit. Taking the sum of  $I(W; F)$  and  $I(P; F)$ , around 33% of the uncertainty remains unexplained. However, from  $I(W, P; F) = 1.500$  bit it can be concluded that



## CHAPTER 6. INTERACTION ANALYSIS

<b>W</b>	<b>P</b>	<b>F</b>	$H(F) = 1.500$ bit
l	l	m	$H(W) = H(P) = 1.000$ bit
l	h	h	$H(\{W, P\}) = H(\{W, F\}) = H(\{P, F\}) = 2.000$ bit
h	l	l	$H(\{W, P, F\}) = 2.000$ bit
h	h	m	$I(W; F) = I(P; F) = 0.500$ bit ( $\approx 33\%$ )
			$I(\{W, P\}; F) = 1.500$ bit (100%)
			<b><math>Q(W; P; F) = 0.500</math> bit (<math>\approx 33\%</math>)</b>

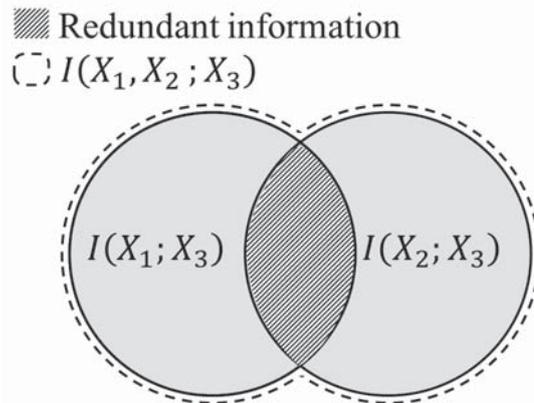
**Figure 6.2:** Summary of the toy example resulting in the measurement of positive interaction information, based on qualitative car data, representing the weight  $W$ , the engine power  $P$  and the acceleration force  $F$  of a car. The respective variables can take discrete values low ( $l$ ), medium ( $m$ ) and high ( $h$ ).

the knowledge about  $W$  and  $P$  together can explain all the uncertainty about the acceleration force  $F$ . The remaining amount of information is due to the synergy of  $W$  and  $P$ , revealed by the result of calculating the interaction information between  $W$ ,  $P$  and  $F$ . With  $Q(W; P; F) = 0.500$  bit, the joint observation of  $W$  and  $P$  conveys the remaining 33% of the uncertainty. Only with the consideration of the interplay between the engine power and the weight a correct prediction of the acceleration force would be feasible.

### 6.4.2 Redundancy

For attributes, where the joint information  $I(\{X_1, X_2\}; X_3)$  is smaller than the sum of  $I(X_1; X_3)$  and  $I(X_2; X_3)$ , e.g.,  $X_1$  and  $X_2$  are redundant in the information they convey about  $X_3$ , as illustrated in Fig. 6.3. In consideration of the conditional information, the information shared between  $I(X_1; X_3|X_2)$  given  $X_2$  is smaller than  $I(X_1; X_3)$ , where  $I(X_1; X_3)$  denotes the information  $X_1$  conveys about  $X_3$  without knowing  $X_2$ . As a consequence, the joint observation of the variables  $X_1$  and  $X_2$  does not provide any additional information about  $X_3$ , and  $I(X_1; X_3)$ ,  $I(X_2; X_3)$  explain equal portions of the uncertainty of  $X_3$ .

The redundancy measured among the three variables indicates that a joint model, which explicitly models the interrelation between  $X_1$  and  $X_2$ , for the purpose of predicting  $X_3$ , is not superior to two individual models, which model the relation between  $X_1$  or  $X_2$  and  $X_3$  respectively. In fact, the two models would further implement two alternative models, which could similarly be used for predicting the same domain of the attribute  $X_3$ . The



**Figure 6.3:** *Diagram illustrating the interrelation between the mutual informations and the joint information, which result in a negative interaction information proportional to the dashed area.*

same holds for deriving sub-components for the design process. Observing only redundancy among sub-components of a system suggests that an independent modeling of the sub-components can be applied. Furthermore, redundant components or variables can provide alternative control parameter to achieve the design and optimization targets. This can get important if modifications of certain parameters are limited due to constraints in the design process.

**Example** Let's reconsider the toy example from above. Neglecting the knowledge about existing physical laws, qualitative data about the acceleration force  $F$ , the engine power  $P$  and the torque  $T$  of an engine are observed. The values of the respective variables are classified into low ( $l$ ), medium ( $m$ ) and high ( $h$ ). The data table with the observations is listed in Fig. 6.4. A simple rule has been applied to generate the data in the table. The rule says that the acceleration force is set to  $l$ ,  $m$  or  $h$ , if the engine power  $P$  and the torque  $T$  is  $l$ ,  $m$  and  $h$ , respectively. Thus, either the knowledge of  $T$  or  $P$  is sufficient to accurately predict  $F$ . The information about  $T$  or  $P$  is completely redundant for the prediction of  $F$ . Fig. 6.4 lists the related results of the different information measures. The uncertainty related to each attribute  $T$ ,  $P$  and  $F$  is measured with  $H(F) = 1.585$  bit. With  $I(T; F) = I(P; F) = 1.585$  bit each variable  $T$  and  $P$  explains 100% of the uncertainty measured for  $F$ . From the calculation of the joint information  $I(T, P; F) = 1.585$  bit, it can be observed that both variables  $T$  and  $P$  together explain not more about  $F$  than  $T$  or  $P$  alone. The negative interaction

<b>T</b>	<b>P</b>	<b>F</b>	$H(T) = H(F) = H(P) = 1.585 \text{ bit}$
l	l	l	$H(\{T, P\}) = H(\{T, F\}) = H(\{P, F\}) = 1.585 \text{ bit}$
m	m	m	$H(\{T, P, F\}) = 1.585 \text{ bit}$
h	h	h	$I(T; F) = I(P; F) = 1.585 \text{ bit (100\%)}$
			$I(\{T, P\}; F) = 1.585 \text{ bit (100\%)}$
			<b><math>Q(\mathbf{T}; \mathbf{P}; \mathbf{F}) = -1.585 \text{ bit } (\approx -100\%)</math></b>

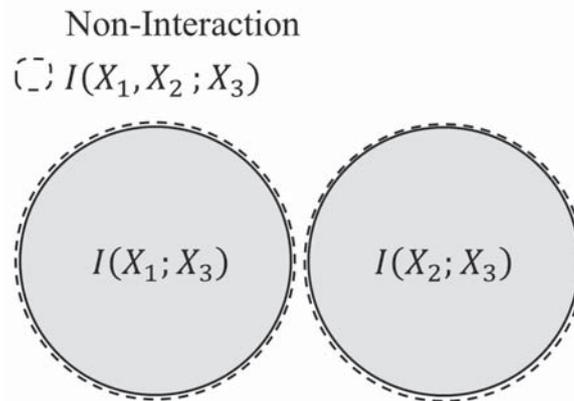
**Figure 6.4:** Summary of the data and information measurements of a toy example resulting in a negative value of interaction information. The table comprised qualitative data about engine variables defining the engine torque  $T$ , the engine power  $P$  and the acceleration force  $F$ , defined through discrete value low ( $l$ ), medium ( $m$ ) and high ( $h$ ).

information  $Q(T; P; F) = -1.585$  denotes that this is due to the redundancy between  $T$  and  $P$  in the context of  $F$ .

### 6.4.3 Non-Interaction

The interaction information can become zero if  $I(\{X_1, X_2\}; X_3)$  equals the sum of  $I(X_1; X_3)$  and  $I(X_2; X_3)$ , as illustrated in Fig. 6.5. Thus, jointly observing the variables can add no information about the uncertainty in the system. In the trivial case  $Q(X_1; X_2; X_3) = 0$ , if at least one variable is independent of the other two. In case that for example  $X_1$  is independent of  $X_2$  and  $X_3$ ,  $H(\{X_1, X_2\}) = H(X_1) + H(X_2)$ ,  $H(\{X_1, X_3\}) = H(X_1) + H(X_3)$  and  $H(\{X_1, X_2, X_3\}) = H(X_1) + H(\{X_2, X_3\})$ . Replacing the entropies in equation (6.5), it can be seen that  $Q(X_1; X_2; X_3)$  vanishes.

Considering the example above, in the case that  $X_1$  is independent to  $X_2$  and  $X_3$ , adding  $X_1$  and modeling its interrelation with the other variables would provide no gain for the prediction of  $X_3$ . As such, if, e.g.,  $I(X_1; X_3) = 1$  and if  $Q(X_1; X_2; X_3) = 0$ , the attribute related to  $X_1$  does not need to be modeled at all. Transferring this concept to the design representation and optimization process, and considering  $X_1$  as a design parameter and  $X_3$  as a performance indicator (e.g., quantifying the drag force),  $X_1$  would provide no control for the improvement of the system performance. As such  $X_1$  can be excluded from the optimization process and can be used as a free parameter to align with other design objectives, e.g., related to aspects of style design.

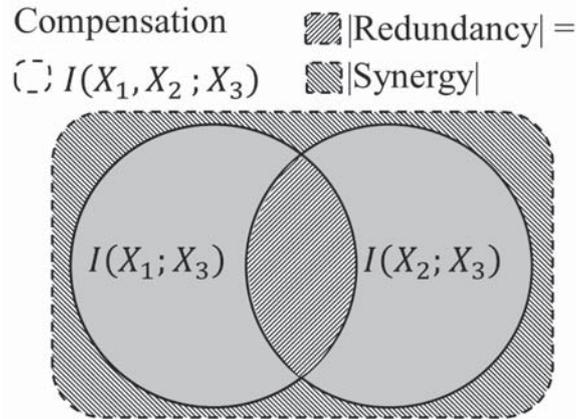


**Figure 6.5:** Illustration of the case where non-interaction is observed among three variables, due to the absence of synergy and redundancy.

### 6.4.4 Compensation

The value of the interaction information does not vanish solely in the total absence of interactions. As discussed in [Krippendorff, 2009a], the interaction information can become zero also in situations where redundancy and synergy cancel each other out, as sketched in Fig. 6.6. Krippendorff argued that  $Q(X_1; X_2; X_3)$  is not a stand-alone measure of redundancy and synergy alone, but measures how the information due to synergy exceeds the redundancy in the system. Within this study this effect has been termed *compensation* or *compensation effect*. The cause is that  $Q(X_1; X_2; X_3)$  represents the expected interaction information over the entire attribute domains. Hence, for some values or value combination of the system attributes the calculation of the interaction information would result in redundancy, while for others a synergy effect by means of a positive interaction information value would be measured. This compensation effect complicates a clear assignment of the measured interaction values to the different classes: synergy, redundancy and non-interaction. Using the value of interaction information for interaction analysis, one always has to consider that the interaction value is a property, which is averaged over the measured domain. Changing the domain slightly can result in a different interaction value. This means that for a decomposition of complex systems into sub-components, the resulting component structure can change dynamically, especially in cases where the scope of design modifications changes, e.g., due to changes in manufacturing constraints.

**Example** Let's reconsider the example from Section 6.4.1, where additional information is revealed due to the synergy of the attributes. In order to show



**Figure 6.6:** Illustration of the compensation effect, when synergy and redundancy cancel each other out. This is the case if the two different dashed areas are of equal size.

the compensation effect, the domain for the three variables acceleration force  $F$ , engine power  $P$  and the weight of the car  $W$  has been extended by adding two new observations. The extended dataset together with the measured information values are listed in Fig. 6.4. The appended observations add redundancy to the system. Finally, this is reflected in the measure of the interaction information, where  $Q(W; P; F)$  drops from  $Q(W; P; F) = 0.500\text{bit}$  in Section 6.4.1 to  $Q(W; P; F) = 0.084\text{bit}$ . The interaction value vanishes, indicating no interaction between the three attributes  $W$ ,  $P$  and  $F$ . However, for the domain, with  $W, P \in \{l, h\}$ , interactions between the attributes would need to be considered, whereas in the domain  $W, P \in \{m\}$  the attributes are redundant. This example provides evidence for the statement of Krippendorff in [Krippendorff, 2009a] that the interaction information value measures how the information due to synergy exceeds redundancy.

## 6.5 Interaction Graphs

For a large number of variables, the analysis of all two- and three-way interactions values can get difficult. *Interaction graphs*, as introduced by [Jakulin and Bratko, 2004a], provide a graphical representation of the most relevant interactions for a selected number of variables. Furthermore, the graphical presentation of the interactions can allow to identify larger interaction patterns between groups of variables, which can hardly be derived from the analysis of the individual information quantities solely. With regard to the engineering design process, the interaction graphs provide engineers means for a visual inspection of the results of the interaction analysis, upon which

<b>W</b>	<b>P</b>	<b>F</b>	$H(F) = 1.252$ bit
l	l	m	$H(W) = H(P) = 1.584$ bit
l	h	h	$H(W, P) = H(W, F) = H(P, F) = 2.252$ bit
h	l	l	$H(W, P, F) = 2.252$ bit
h	h	m	$I(W; F) = I(P; F) = 0.584$ bit ( $\approx 47\%$ )
m	m	m	$I(W, P; F) = 1.252$ bit ( $\approx 79\%$ )
m	m	m	<b><math>Q(W; P; F) = 0.084</math> bit (<math>\approx 7\%</math>)</b>

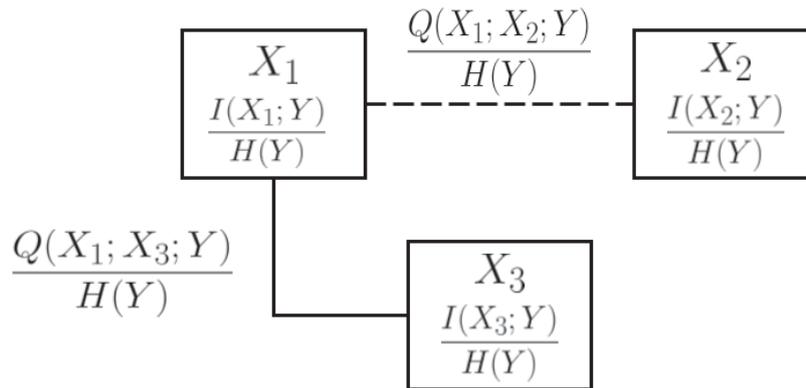
**Figure 6.7:** Example, illustrating the compensation effect, where information due to synergy and redundancy cancel each other out. The example comprises discrete observations for low (*l*), medium (*m*) and high (*h*) values of engine power *P*, car weight *W* and acceleration force *F*.

new hypothesis about the interrelations between design parts can be derived and tested. Therefore, interaction graphs can be a valuable support for engineers to efficiently decompose the design of a technical system into sub-components and processes.

In [Jakulin and Bratko, 2004a], the authors introduced two types of graphical representations, namely supervised and un-supervised graphs. Un-supervised graphs show general interaction patterns among all attributes, while supervised graphs visualize interactions in the context of an a priori selected supervision or class variable. In this work, supervised graphs have been adopted for the purpose of visualizing and analyzing design interactions, in the context of a particular performance measure defining the design quality.

As an example, Fig. 6.8 depicts the structure of a supervised graph, which contains information about two- and three-way interactions among three design variables  $\mathcal{X} = \{X_1, X_2, X_3\}$ . The performance variable is denoted by  $\mathcal{Y} = \{Y\}$ . The design variables in  $\mathcal{X}$  relate to adjustable parameters, which define the design configuration (e.g., geometry), whereas the variable *Y* relates to an measured attribute that defines the design performance (e.g., drag force).

In the example graph, *Y* is selected as supervision attribute. Thereafter, all information quantities that are shown in the graph are normalized by the entropy  $H(Y)$ , since the overall target is to identify those parameter and parameter interactions, which add to the reduction of the uncertainty about the prediction of *Y*.



**Figure 6.8:** *Visualization of the information graph for two- and three-way interactions.*

Each graph consists of knots and edges connecting them. A knot is labeled with the variable name and the value of the relative mutual information (two-way interaction), e.g.,  $I(X_1, Y)/H(Y)$ , whereas edges denote the relative interaction information (three-way interaction), e.g.,  $Q(X_1; X_2; Y)/H(Y)$ . In order to clearly distinguish between synergy and redundancy, connections representing positive interaction values are depicted as straight lines, whereas connections representing negative interaction values are depicted as dashed lines between knots in the graph. For a more compact graph representation, the strength of the interactions might be represented by the line thickness instead of adding a label with the relative interaction value to each edge. In the case that the interaction value vanishes, either because of a non-interaction or due to the compensation effect, connections have been removed from the graph to improve its readability.

For the visualization of higher order interactions, the interaction graph in the form of Fig. 6.8 can become inappropriate. If higher order interactions need to be considered, block diagrams as used by Lüdke et al. in [Lüdtké et al., 2008] are a valuable alternative, but allowing the visualization of the information balance for individual dependent variables only.

## 6.6 Utilizing Interaction Information

---

The results from an interaction analysis can be utilized for versatile purposes within engineering design and optimization processes. For a given decomposition and design of a technical system, interaction analysis provides the means to verify and revise the decomposition based on the estimated functional interdependencies. Given new data or changing objectives, the decomposition of a technical system can and should continuously be tested. Beside

verifying existing system decompositions, the interaction analysis allows to provide support for an initial problem decomposition, e.g., in cases where no a priori knowledge about a technical system is given. Concerning computational design optimizations, the results from the interaction analysis can be utilized to tune a working object representation, targeting the setup of an efficient initial representation, or the change of the object representation online during individual design optimization runs.

Originally, Jakulin et al. in [Jakulin and Bratko, 2003; Jakulin et al., 2003] utilized the results from the interaction analysis to build up efficient mathematical models for classification tasks. Hence, the results from the interaction analysis can as well be utilized to define the structure of mathematical models, which approximate the mapping between design parameters and performance numbers efficiently. Applying such models as surrogates for expensive performance evaluations can significantly speed up computational design optimization processes, e.g., see [Jin, 2011]. Equally, such models can be applied in engineering design processes to predict the design performance, e.g., in early phases of a design process [Fukushige et al., 2012], and as such supporting and speeding up decision making.

## **6.7 Examples**

---

The correct interpretation of the information quantities is essential to draw meaningful conclusions about the interrelations between parameters. Two simple examples are outlined, where aspects regarding the normalization of the information measures as well as the compensation property of the 3-way interaction value are studied.

### **6.7.1 Information Normalization**

As stated earlier, the Shannon entropy of a discrete uniformly distributed random variable increases with the number of discrete states or symbols the variable can take. This states an important property of the Shannon entropy for analyzing the capacity and information exchange for communication channels. However, for the quantification of the importance of parameter and parameter interactions with respect to a performance variable, the absolute amount of information or uncertainty is of minor relevance. Rather, their relative contribution for the explanation of the uncertainty of the performance variable is of interest. Therefore, the information quantities are normalized by the entropy of the performance variable. In the following examples, the sensitivity of the normalized information measures to the number of discrete states is investigated.

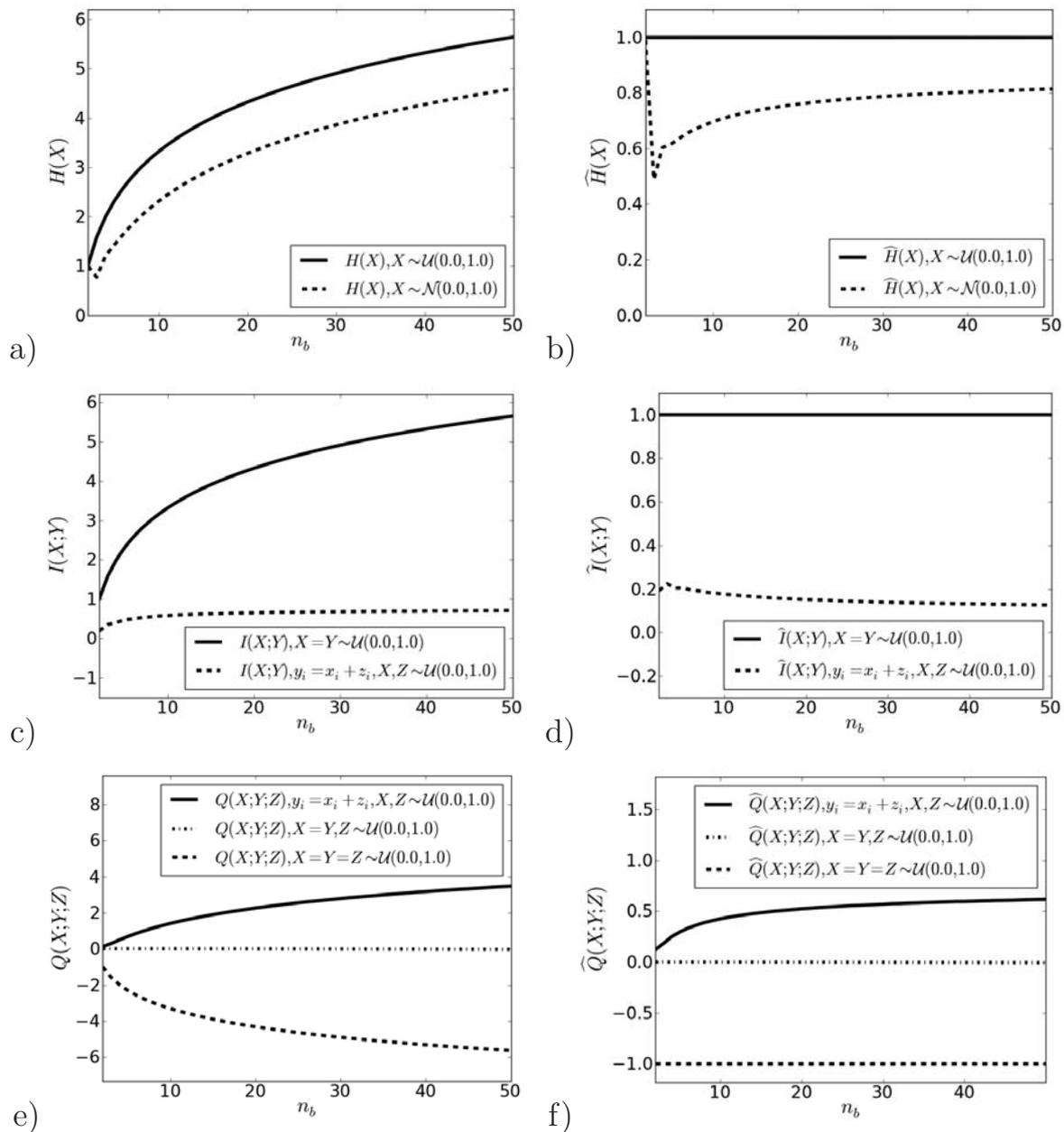
## CHAPTER 6. INTERACTION ANALYSIS

---

Considering a discrete random variable  $X$ , which has been derived from the discretization of a continuous random variable  $X_C$  into  $n_b$  intervals of equal length, Figs. 6.9 a) and b) reveal the dependency of the Shannon entropy on the number of discrete intervals, with  $n_b \in [2, 50]$ . The two plots a) and b) show the values of the entropy  $H(X)$  and the normalized entropy  $\hat{H}(X) = H(X)/\log_2(n_b)$ , respectively, where  $\log_2(n_b)$  defines the maximum entropy that can be measured for the discrete random variable. The values of the random variable  $X_C$  are either drawn from a uniform or a normal distribution, so that  $X_C \sim \mathcal{U}(0, 1)$  (solid line) or  $X_C \sim \mathcal{N}(0, 1)$  (dashed line). For simplicity the following notation is equally used for the discrete random variable:  $X \sim \mathcal{U}(0, 1)$  and  $X \sim \mathcal{N}(0, 1)$ . In all experiments the sample size for  $X_C$  is constant with  $n = 50000$ . Since the maximum value of the entropy is  $\log_2(n_b)$ , the normalized entropy for a uniform distributed random variable equals  $\hat{H}(X) = 1$ , as shown in Fig. 6.9 b). For non-uniform distributed random numbers, like  $X \sim \mathcal{N}(0, 1)$ , not only does  $H(X)$  change with increasing  $n_b$ , but also does  $\hat{H}(X)$ . It can be seen that for smaller number of discrete values the relative entropy is underestimated. In [Miller, 1955], the authors studied such biases in the estimation of the entropy in more detail.

As  $H(X)$  increases with the number of discrete states, the amount of information that can be transferred between  $X$  and another variable  $Y$  increases too. Taking  $Y$  as a uniform distributed random variable, the maximum information that  $X$  can convey about  $Y$  is the amount of its uncertainty  $H(Y) = \log_2(n_b)$ . In the trivial case this occurs if the second variable  $X$  equals  $Y$ , so that  $I(X; Y = X) = H(X)$ , see Fig. 6.9 c). Thus, the value of the normalized mutual information becomes  $\hat{I}(X; Y) = I(X; Y)/H(Y) = 1.0$ , as shown in Fig. 6.9 d) (solid line).

Of more practical relevance is the case in which  $Y$  is determined by two uniformly distributed random variables  $X$  and  $Z$ , following a deterministic function  $y_i = x_i + z_i$ . If only  $X$  and  $Y$  are observed, with  $\hat{I}(X; Y) \approx 0.15$ ,  $X$  can explain only a small amount of the uncertainty in  $Y$  or vice versa, shown in Fig. 6.9 d) (dashed line). A large amount of uncertainty is explained by the joint observation of  $X$  and  $Z$ , as indicated by the positive value of the interaction information  $\hat{Q}(X; Y; Z)$  or  $Q(X; Y; Z)$ , see Figs. 6.9 e) and f) (solid line). For reference a negative interaction information is depicted in the graph (dashed line), which is the result of calculating  $\hat{Q}(X; Y; Z)$  and  $Q(X; Y; Z)$ , with  $X = Y$ . While the normalized negative interaction information value remains constant with increasing  $n_b$ , the value of the positive interaction information increases logarithmically.



**Figure 6.9:** Results from analyzing the sensitivity of the information measures to the number of discrete intervals are shown, where a) and b) relate to the Shannon entropy, c) and d) to the mutual information and e) and f) the measure of interaction information, both to the normalized (right) as well as the un-normalized measures (left), respectively.

This analysis shows that the values of the normalized information measures do in principal depend the number of discretization intervals  $n_b$ .

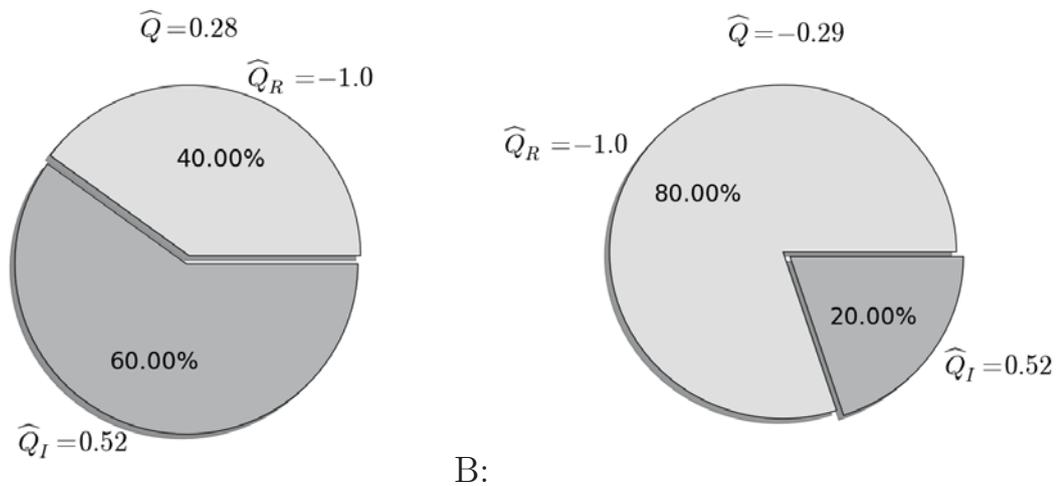
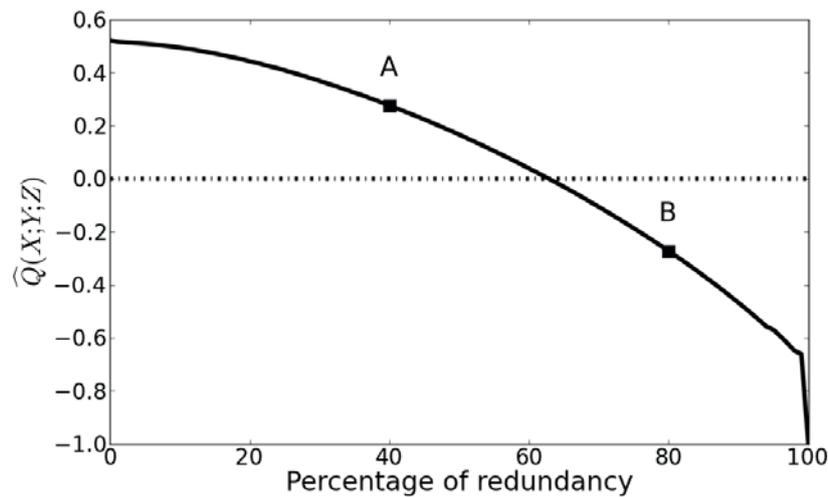
### 6.7.2 Compensation of Redundancy and Synergy

The sign of  $Q(X; Y; Z)$  can be used as indicator of synergy and redundancy among the variables  $X$ ,  $Y$  and  $Z$ . If a variable  $X$  adds exactly the same information to  $Y$  as a third variable  $Z$ ,  $X$  or  $Z$  are completely redundant in the context of  $Y$  and  $Q(X; Y; Z)$  is minimal. Trivially,  $Q(X; Y; Z)$  is minimal if  $X = Y$  and  $Y = Z$ , as shown in Figs. 6.9 e) and f) (dashed line). In such cases, if  $X$  is drawn from a uniform distribution then  $Q(X; Y; Z) = -\log_2(n_b)$ , where  $n_b$  defines the number of discrete states  $X$  can take.

In contrast to the case of maximum redundancy, in the example where  $y_i = x_i + z_i$  and  $X, Z \sim \mathcal{N}(0.0, 1.0)$ ,  $Q(X; Y; Z)$  quantifies the synergy effect of  $X$  and  $Z$  on  $Y$ , with  $Q(X; Y; Z) > 0$ . If in average no additional information about  $Y$  can be inferred from the synergy of  $X$  and  $Z$  then  $Q(X; Y; Z) = 0$ . For example, this is the case, when  $Y$  is independent from  $Z$ , as depicted in Figs. 6.9 e) and f) (dotted line), where  $X = Y$  and  $X, Z \sim U(0.0, 1.0)$ . However,  $Q(X; Y; Z)$  can be zero also in cases where the amount of information due to the synergy and the amount of redundant information cancel each other out. An instructive example of this effect is shown in Fig. 6.10, where  $Q(X; Y; Z)$  has been calculated for increasing portions of induced redundancy among three random variables  $X$ ,  $Y$  and  $Z$ . Starting with 0% redundancy the parameter setting equals the example, where  $X, Z \sim U(0.0, 1.0)$  and  $y_i = x_i + z_i$ . In this setup,  $Q(X; Y; Z)$  or the respective normalized variant  $\hat{Q}(X; Y; Z)$  gets maximal, and about 53% of the uncertainty in  $Y$  are explained by the interaction between  $X$  and  $Z$ . Gradually, portions of the data samples  $x_i$ ,  $y_i$  and  $z_i$  are replaced so that  $z_i = y_i = x_i$ , thus increasing the overall amount of redundancy in the data. As result,  $\hat{Q}(X; Y; Z)$  gradually decreases from high interaction to maximum redundancy, from  $\hat{Q}(X; Y; Z) = 0.53$  to  $\hat{Q}(X; Y; Z) = -1.00$ . The interaction value becomes  $\hat{Q}(X; Y; Z) = 0.00$ , when about 62% of the samples are redundant.

Two concrete datasets  $A$  and  $B$  are explicitly shown in Fig. 6.10, where in dataset  $A$  40% and in dataset  $B$  80% of the observations add to redundancy. Calculating the 3-way interaction results in  $\hat{Q}(X; Y; Z) = 0.28$  and  $\hat{Q}(X; Y; Z) = -0.29$  for dataset  $A$  and  $B$ , respectively. Each dataset  $A$  and  $B$  is split into two subsets, where each subset contains either all observations that add to redundancy or those that add to synergy. Quantifying the 3-way interaction for each subset independently,  $Q(X; Y; Z)$  remains at its maximum and minimum value, respectively.

Redundancy and synergy are a property of individual observations in the data and the interaction information is quantifying the balance between the observations which either add to redundancy or synergy. For applying inter-



**Figure 6.10:** Analysis of the dependency of  $Q(X;Y;Z)$  from the fraction of redundancy induced to an interacting structure. The analysis shows that  $Q(X;Y;Z)$  can get zero in cases where the amount of information from the interaction and the amount of redundant information cancels each other out.

action analysis to data of unknown characteristics, it can be of importance to discover which observations add to the redundancy or synergy subset. However, identifying the two subsets from data is a non-trivial task and goes beyond the scope of this thesis.

## 6.8 Application to Passenger Car Design

---

In this section, the interaction analysis is applied to passenger car design data, with the target to discover the most relevant interactions between distant surface areas. Therefore, exemplary, the two- and three-way interactions are calculated based on dataset 4 (Section 3.5). The dataset covers the surface and performance data of the designs from the two optimization processes, where different representations  $\mathcal{R}_A$  and  $\mathcal{R}_B$  have been used. Details on the optimization processes are presented in Section 2.4.4. According to the surface mesh representation and the performance numbers, the displacements and performance differences have been calculated. The basis of the interaction analysis is the displacement data linked to the reduced design features  $A_0$  to  $A_5$ , which has been the result of the automatic feature reduction step in Chapter 5. The position of the respective vertices on the baseline car shape are shown in Fig. 4.8. In addition to the displacements of the reduced feature set, the performance differences in drag  $\phi_{FD}$ , rear lift  $\phi_{FLR}$  and the combined fitness  $\phi_F$  (equation (5.9)) are used for the interaction analysis. Each of the variables is discretized into  $n_b = 10$  bins of equal length, based on which the probabilities are derived by calculating the relative frequencies. The estimation of the joint probabilities is carried out accordingly. The transformation results in the following discrete random variables:  $\mathcal{X} = \{A_0, \dots, A_5\}$  and  $\mathcal{Y} = \{Y\}$ , with  $Y \in \{\phi_{FD}, \phi_{FLR}, \phi_F\}$ .

Interaction graphs from Section 6.5) are used to visualize and analyze the calculated information measures. For each of the performance numbers the interactions are analyzed individually. The information measures are normalized by the respective entropy, i.e.,

- $H(\phi_{FD}) = 2.98,$
- $H(\phi_{FLR}) = 2.86,$
- $H(\phi_F) = 2.73,$

which quantify the uncertainty in predicting the performance values. The resulting interaction graphs are depicted in Figs. 6.11 a) to c). Each node in the graphs is representing a discrete variable of one of the parameters in  $\mathcal{X}$ .

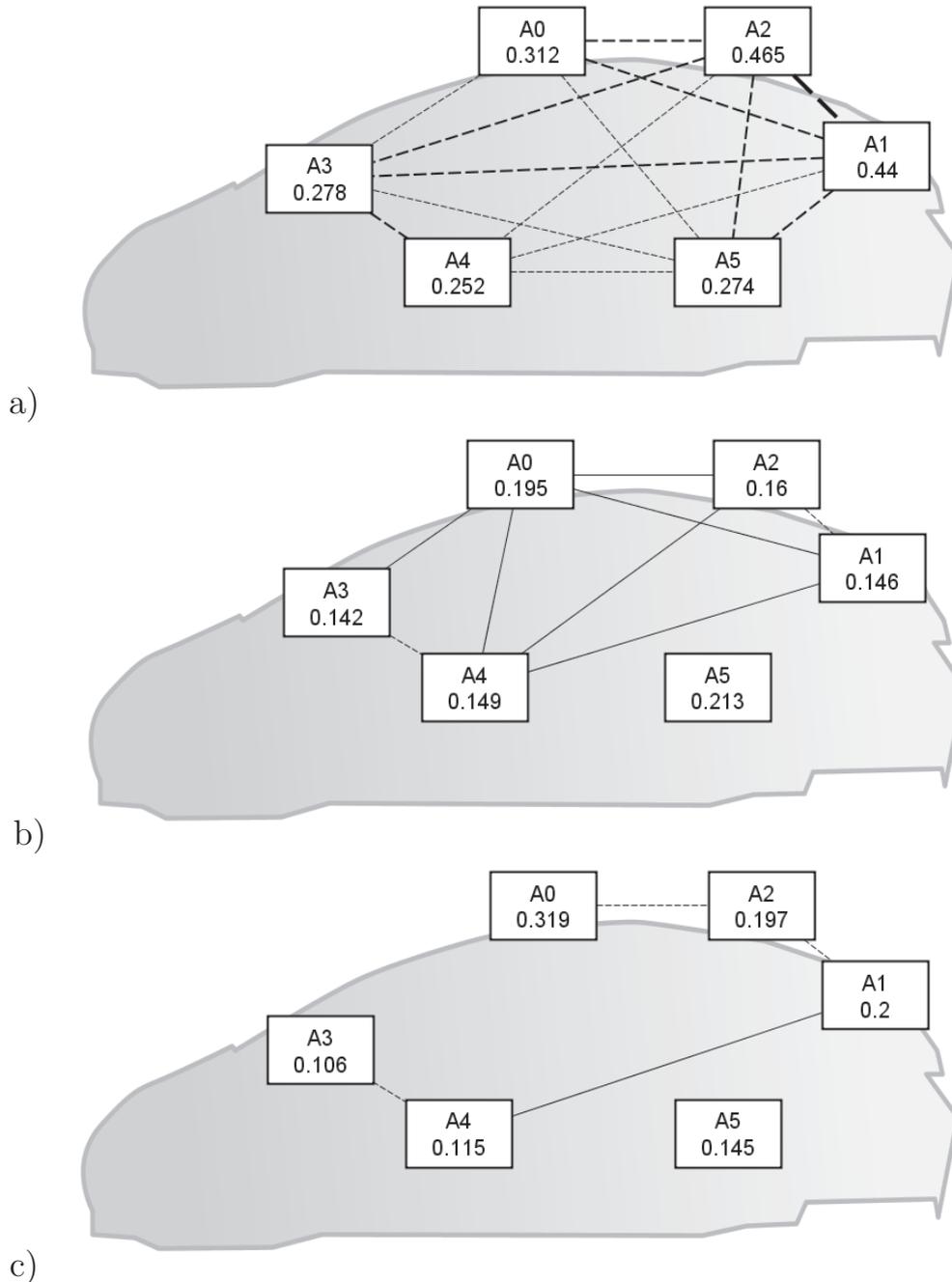
## 6.8. APPLICATION TO PASSENGER CAR DESIGN

---

In addition to the interaction graphs the cross-section contour of the baseline car is visualized to make the link between the parameters and the respective surface areas. Therefore, the nodes are roughly drawn at the respective position of the parameters on the baseline surface. Beside the parameter name, the caption in each node labels the value of the relative two-way interaction, by means of the relative mutual information between the parameter and the performance values, e.g.,  $I(A0; \phi_{F_D})/H(\phi_{F_D})$ . The edges between two nodes define the interaction between two parameters with respect to the performance, e.g.,  $Q(A0; A1; \phi_{F_D})/H(\phi_{F_D})$ . Solid lines reveal synergy and dashed lines redundant interrelations, respectively. The thickness of each line corresponds to the strength of the interaction. Edges with an relative interaction value below 0.05 are classified as non-interaction and are not visualized in the graph.

Fig. 6.11 a) depicts the interaction graph of the surface parameters in the context of the drag force. From the analysis of the two-way interactions the parameter  $A2$  is identified as the most important one. With  $I(A2; \phi_{F_D})/H(\phi_{F_D}) = 0.465$ , the variations of  $A2$  convey already a large amount of information about the drag force measurements  $\phi_{F_D}$ , and explain around 47% of its uncertainty. Analyzing the three-way interactions of all parameter combinations reveals redundancy among the design parameters, meaning that all values of the calculated relative interaction information are below 0.05. This indicates that on average all design parts could be independently tuned for an optimal drag value. The risk is expected to be low that a subsequent combination of the optimization results will lead to unexpected measures of  $F_D$ . The amount of redundancy is especially high concerning the interaction between parameters  $A1$ ,  $A2$  and  $\phi_{F_D}$ , where  $A1$  is representing displacements at the roof and  $A2$  representing displacements along the rear window of the car surface. Thus, e.g.,  $A1$  provides an alternative handle to the change of the drag force, e.g., if modifications of  $A2$  are restricted or vice versa. Such restriction can be given, e.g., due to manufacturing or aesthetic constraints.

Subsequently, the interactions among the parameters in the context of rear lift force  $\phi_{F_{LR}}$  are investigated, as shown in Fig. 6.11 b). Analyzing the two-way interactions, in contrast to the interaction graph relating to the drag force, parameter  $A5$  instead of  $A2$  has the highest influence on the rear lift force. Thus, the rear lift force is especially sensitive to changes of  $A5$ . Concerning the analysis of the three-way interactions, no significant interactions of parameter  $A5$  with other parameters are observed. Thus, for the optimization or modeling of the rear lift force the area represented by  $A5$  can be considered as independent with respect to other areas of the car. Further analysis are required to exclude that the absence of interaction is



**Figure 6.11:** Illustration of the interaction graphs concerning a)  $\phi_{FD}$ , b)  $\phi_{FLR}$  and c)  $\phi_F$ . Each node shows one design parameter and the edges between them show their interactions, where solid lines reveal synergy and dashed lines redundancy among them.

due to the compensation of redundancy and synergy. In contrast to A5, e.g., A0 to A4 have to be considered jointly, when targeting the optimization or modeling of the roof or the front side of the car shape.



## 6.8. APPLICATION TO PASSENGER CAR DESIGN

---

In order to analyze the design parameter interrelations concerning the trade-off between the drag force and the rear-lift force, the interaction analysis with respect to the combined fitness  $\phi_F$  is carried out. The respective interaction graph is illustrated in Fig. 6.11 c). The graph denotes the importance and interrelation between surface areas, which are relevant, e.g., when targeting the design of an optimal car shape with respect to both criteria. The interaction analysis identified  $A_0$ ,  $A_1$  and  $A_2$ , which represent areas along the roof, to be the most relevant parameters. Interestingly, a relevant interaction is discovered between  $A_1$  and  $A_4$ , representing modifications at the back and the front side of the car. The consideration of such interactions in the design process might per se not be obvious to the engineer, but is of high relevance.

These examples show that the interaction analysis can be exploited as a helpful tool to identify and characterize relevant interrelations between design parameters with respect to the design performance. Depending on the considered performance criteria, different interaction patterns can be of relevance. Thus, depending on the objectives of a design process, different problem decompositions need to be deployed, which is often not considered in practical applications. Furthermore, as shown by the example of studying  $\phi_{FD}$ , redundancy among parameters can reveal alternative design strategies, e.g., in situations when the design of an object is constrained by manufacturing, aesthetic or other constraints.

## 6.9 Interim Summary

---

In this chapter an information theoretic attempt for analyzing the interactions between distant design areas has been studied. It has been demonstrated that the interaction information, an extension of the mutual information to multiple variables, provides computational means to reveal functional interrelated design areas. Thus, it represents a valuable tool to support the decomposition of complex design tasks into subcomponents. Such a decomposition is often the only way to model and optimize complex systems. The results from the analysis of the passenger car design data have shown, that for different objectives, relating to the design performance, the interrelations can change. While in practice the decomposition of complex systems often remains fixed over the course of the design process, interaction analysis allows to review and revise the construction of subcomponents, especially in cases where the objectives for the design process change.



## CHAPTER 6. INTERACTION ANALYSIS

---

An additional focus of this chapter has been the interpretation of the measure of interaction information and its consequences for the design decomposition and optimization. The measure of interaction information has the unique property that it provides a means for quantifying synergy and redundancy among design areas at the same time. However, as has been pointed out, care has to be taken with the interpretation of the information measures, especially in situations where effects of redundancy and synergy cancel each other out.



# 7

## Summary and Outlook

With the increased application of computer aided engineering technologies for system design, the analysis of the emerging design data has become an important activity for improving future design processes. Therein, modern data mining technologies can provide powerful tools for supporting the analysis and decision making process. The huge variability in computational design representations usually poses a big challenge to the computational analysis and modeling of all of the accumulated design data. The technologies and ideas for shape mining, presented in this thesis, provide means for a holistic analysis of designs, which are defined by its geometrical surface.

### 7.1 Summary

---

The thesis has motivated the need for a holistic analysis of engineering design data for an efficient integrative design process, defined in Chapter 2. It has been emphasized that the definition of a unified design representation is a pre-requisite to provide a holistic view on the data, and to facilitate the implementation of a holistic data mining process. A holistic view on the design data allows an integrated exchange of domain knowledge and that design decisions can be made in consideration of the entire design configuration. Both are means to increase the efficiency of serial and concurrent design processes.

Related to the analysis of design data in the domain of aerodynamic and structural design, this thesis has defined unstructured surface meshes as a unified representation of object shapes in Chapter 3. This representation facilitates the combination of information from different design processes into one data mining approach, i.e., the shape mining process. The shape mining process is exploiting data mining methods to extract valuable knowledge from shape and performance data, including methods for sensitivity analysis, design concept retrieval and interaction analysis.



## CHAPTER 7. SUMMARY AND OUTLOOK

---

Ideas for shape mining have been studied based on a realistic application to the design of the outer shape of a passenger car. As described in Chapter 2, different optimization and sampling strategies, together with different shape representations have been used to generate a heterogeneous design dataset, where the performance of the designs has been evaluated based on the results of high fidelity CFD simulations. The definition of a unified shape representation has allowed to model the relationship between shape deformations and performance changes of the car with respect to all designs in the datasets. However, the high dimensionality of the shape representation has prevented a naive usage of enhanced modeling technologies.

The definition of local shape features and the evaluation of the feature disparities by means of vertex displacements, has allowed a detailed study of the differences between surface meshes, see Chapter 3. Those differences define the cause for changes in the design performance. This thesis focused on the analysis of local vertex displacements for the investigation of shape deformations relative to a specific baseline design. The estimation of the displacements has required the identification of corresponding vertices. As shown by the analysis of the passenger car design data, the computation and visualization of displacement statistics provides valuable information for the investigation of individual design solutions, the identification of weakly explored design regions and the evaluation of the course of design processes.

Besides the actual design configuration, engineering design data also includes performance numbers, which denote the quality of each design. In order to analyze the effect of design variations on the design performance, in Chapter 4 methods for sensitivity analysis have been studied. In addition to linear correlation techniques like the Pearson correlation coefficient, the mutual information index and the novel robust mutual information measure have been investigated. Based on the passenger car design data, it has been shown that, in contrast to the mutual information index, the robust mutual information measure provides reliable estimates even for a smaller number of designs. Furthermore, the results from the sensitivity analysis denote areas of the car shapes, which influence, e.g., the drag or the rear lift of the car. Beyond the study of methods for sensitivity analysis, a new attempt utilizing the K-nearest neighbor search algorithm has shown to provide control over the locality of the sensitivity estimates.

Exploiting the geodesic surface distance and the sensitivity information, in Chapter 4 a new approach has been introduced to automatically derive larger sensitive areas from the design surface. Therefore, vertices of the surface mesh have been grouped into distinct surface areas based on the sensitivity information and their geodesic distance. The displacement data of the vertices closest to the center of the surface areas makes up a low dimensional data representation. This dimensionality reduction has been

an important step in the shape mining process, because it facilitates the modeling of the interrelation between distant design areas. The extraction of sensitive areas, e.g., facilitates the modeling of passenger car design concepts.

Chapter 5 has introduced a class of techniques for design concept retrieval, with the target to abstract a manageable set of concepts from a huge number of designs, which can easily be communicated to the design engineer. The approach for the retrieval, description and evaluation of design concepts has been generalized independent of a specific modeling technique. For an additional filtering of the set of concepts, a new measure has been introduced to evaluate design concepts based on the estimation of their utility. The utility measure combines a measure of correlation between the antecedent and the consequent of an rule, representing the concept, with the hypervolume indicator, which quantifies compliance with a priori defined design objectives. Thus, the new measure allows to rank concepts according to the formulation of the engineer's objectives. For example, using self-organizing maps together with the utility measure has resulted in a limited set of three relevant passenger car design concepts, which are most relevant to minimize the drag and the rear lift force of the car shape.

In order to reveal functional interrelated design areas, in Chapter 6 an information theoretic attempt has been studied, aiming at exploiting this information to decompose complex design tasks into independent subcomponents. For this purpose, the measure of interaction information, an extension of the mutual information to multiple variables, has been studied. It provides a measure to evaluate the relevance of interrelations, which can be classified into synergy and redundant relations among design areas with respect to the considered performance number. By applying interaction analysis to the passenger car design data, and utilizing interaction graphs for the visualization of the resulting information, it has been shown that the interaction analysis can provide valuable information about the interrelation between design areas, which might not be obvious to the engineers.

## 7.2 Outlook

---

Primarily, the thesis has investigated shape mining methodologies for knowledge extraction, while leaving the aspect of utilizing the knowledge aside. The most interesting follow up research ideas center around two questions:

- Does the utilization of the extracted design knowledge speed up the synthesis process of new designs or improve the final design quality?



## CHAPTER 7. SUMMARY AND OUTLOOK

---

- How can the extracted knowledge be utilized to improve computational design optimization algorithms to speed up the search process?

First ideas for utilizing the extracted knowledge have already been given throughout the thesis. However, a detailed study and the development of elaborated processes and technologies remains future work. One concrete approach is concerned with the usage of sensitivity information for setting up the design representation for an efficient optimization process, as introduced in [Graening et al., 2012]. However, using sensitivity information only can limit the search process, and one can not expect to discover conceptually different design solutions. Thus, information like the variance of surface deformations can encourage a broader exploration of design solutions. Managing the trade-off between exploration and exploitation in the design process using sensitivity or variance information seems an interesting research topic.

Beyond that, the thesis has covered and combined technologies from various different disciplines, which have been integrated into the shape mining process. For example technologies from computational geometry, machine learning, data mining, information theory, optimization, aerodynamics and others have been combined. As a consequence, many opportunities exist to extend this research work. In the following, a brief summary of possible enhancements is given.

The evaluation of the local surface differences between two different designs requires the identification of corresponding vertices. For structural changes of the design, the considered algorithms can result in wrong estimates of corresponding vertices. Depending on the expected shape differences, alternative methods, which consider structural changes need to be considered. However, more sophisticated attempts, like the one proposed by Chui and Rangarjan [Chui and Rangarjan, 2000], often go along with an significant increase in computation time.

The identification of sensitive areas requires the application of clustering techniques to group vertices on the design surface. The application of standard clustering techniques from machine learning do not take any symmetries of points into account, which often requires a post-processing of the clustering results before the interrelations between a reduced set of design features and the design quality can be carried out efficiently. Future work, may consider the research on sophisticated clustering techniques that take symmetries into account.

Methods for information theoretic interaction analysis provide an interesting approach to decompose complex designs into manageable sub-components. A comprehensive treatment of the underlying theory has not been in the scope of the thesis. In future studies it would be of interest, e.g., to exten-



sively investigate the meaning of higher-order interactions, or to investigate the actual effect of the design decomposition on the overall design process.

While experienced engineers can accurately judge the design interactions based on individual objectives, the understanding of interactions for multiple objectives is often unclear. The interaction analysis could provide a computer aided approach to uncover those interrelations, but requires further research into that direction.





# A

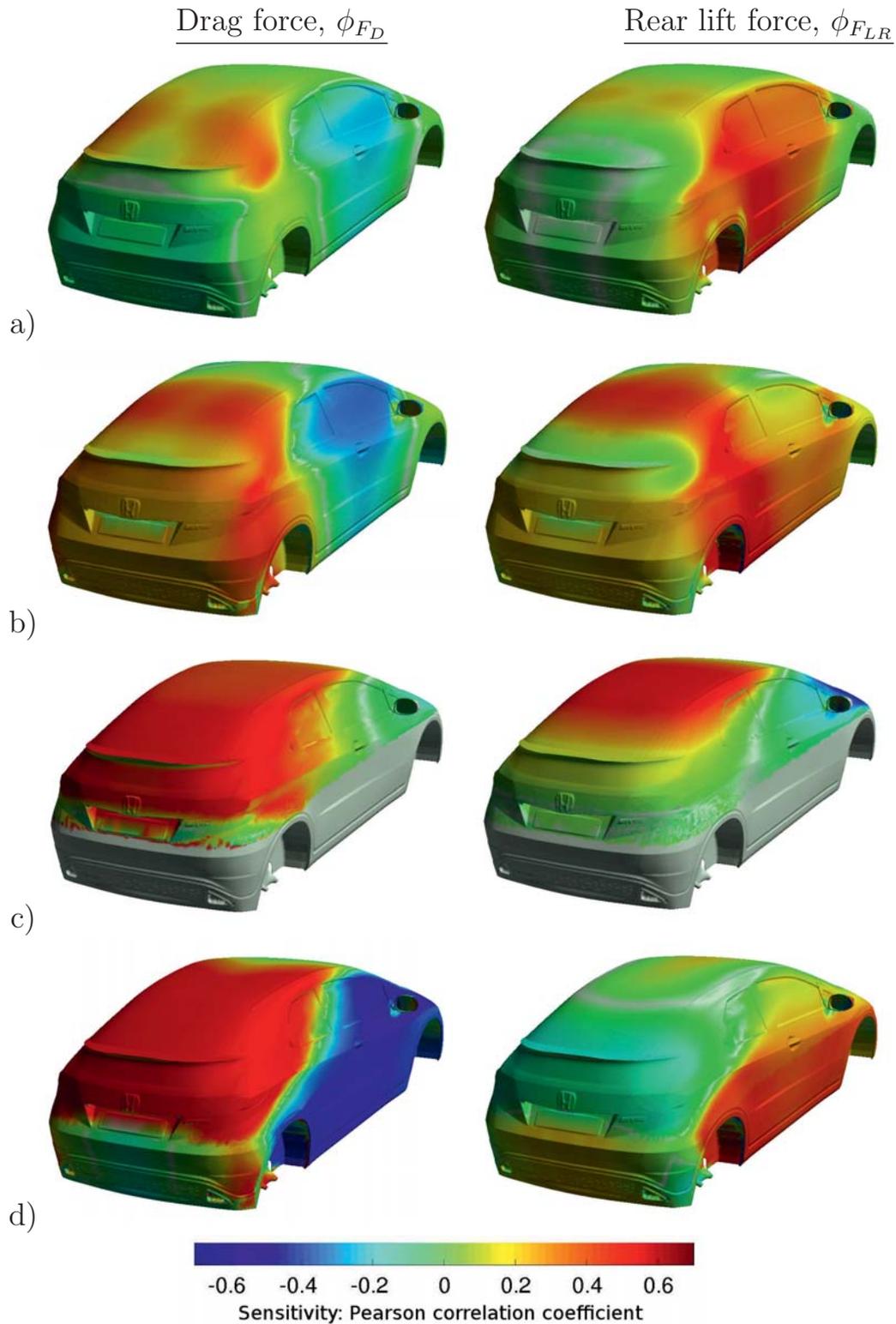
## Design Sensitivity Analysis

### **A.1 Direction of performance improvement**

---

Figs. A.1 visualizes the sensitivities calculated using the Pearson correlation coefficient. For completeness, the sensitivities around the rear of the reference mesh are shown. The results are depicted for different datasets 1 to 4 and for two different performance difference indicators,  $\phi_{FD}$  and  $\phi_{FLR}$ .

## APPENDIX A. DESIGN SENSITIVITY ANALYSIS



**Figure A.1:** Results of the sensitivity analysis based on the calculation of the Pearson correlation coefficient calculated for different passenger car design datasets 1 to 4, shown in row a) to d). On the left side the sensitivities with respect to  $\phi_{FD}$  and on the right side the sensitivities with respect to  $\phi_{FLR}$  are depicted.



# B

## Related Publications

1. **L. Graening**, B. Sendhoff: *Shape Mining: A Holistic Data Mining Approach for Engineering Design Synthesis*, submitted to *Advanced Engineering Informatics*, Elsevier, 2013.
2. T. Fukushige, T. Ariyoshi, Y. Uda, Y. Okuma, T. Okabe, K. Yamamoto, **L. Graening**, S. Menzel, M. Olhofer: *Development of Real-time Aerodynamic Design System*, In *Honda R&D Technical Review* 24, 2012.
3. **L. Graening**, S. Menzel, M. Hasenjaeger, T. Bihrer, M. Olhofer, B. Sendhoff: *Knowledge Extraction from Aerodynamic Design Data and its Application to 3D Turbine Blade Geometries*, In *Journal of Mathematical Modelling and Algorithms*, 7(4), 2008.
4. **L. Graening**, S. Menzel, T. Ramsay, B. Sendhoff: *Application of Sensitivity Analysis for an Improved Representation in Evolutionary Design Optimization*, In *Proceedings of the IEEE International Conference on Genetic and Evolutionary Computing (ICGEC)*, 2012.
5. M. Rath, **L. Graening**: *Modeling Design and Flow Feature Interactions for Automotive Synthesis*, In *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, 2011.
6. **L. Graening**: *Knowledge Extraction from Aerodynamic Design Data and its Application to 3D Turbine Blade Geometries*, In *International Workshop on Machine Learning for Aerospace*, Marseille, France, 2009, ([http://videlectures.net/lars\\_graening](http://videlectures.net/lars_graening)).
7. **L. Graening**, M. Olhofer, B. Sendhoff: *Interaction Detection in Aerodynamic Design Data*, In *Proceedings of the 10th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, 2009.



## APPENDIX B. RELATED PUBLICATIONS

---

8. **L. Graening**, M. Olhofer, B. Sendhoff: *Knowledge Extraction from Unstructured Surface Meshes*, In Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), 2007.



- N. Abramson. *Information Theory and Coding*. McGraw-Hill, New York, 1963.
- M. J. Aftosmis, M. Delanaye, and R. Haimes. Automatic generation of cfd-ready surface triangulations from cad geometry. *AIAA Paper 99-0776*, 1999.
- M. Alexa. Recent advances in mesh morphing. *Computer Graphics Forum*, 21(2):173–197, 2002.
- J. Ascough, G. Thimoty, M. Liwang, and A. Lajpat. Key criteria and selection of sensitivity analysis methods applied to natural resource models. In *International Congress on Modeling and Simulation Proceedings*, Salt Lake City, UT, November 2005.
- P. J. Besl and R. C. Jain. Invariant surface characteristics for 3d object recognition in range images. *Comput. Vision Graph. Image Process.*, 33: 33–80, January 1986.
- P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:239–256, February 1992.
- H. G. Beyer and H. P. Schwefel. Evolution strategies - a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.
- M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*. AK Peters, 2010. ISBN 978-1-56881-426-1.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
- S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD International Conference on Management of Data*, 1997.
- A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008.
- D. Chen-shi and W. Guo-zhao. Curvatures estimation on triangular mesh. *Journal of Zhejiang University Science*, 6A:128–136, 2005.
- K. Chiba, S. Jeong, S. Obayashi, and H. Morino. Data mining for multidisciplinary design space of regional-jet wing. *IEEE Congress on Evolutionary Computation*, 3:2333–2340, 2005.



## BIBLIOGRAPHY

---

- H. Chui and A. Rangarjan. A new algorithm for non-rigid point matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 44–51, Hilton Head Island, SC, US, June 2000.
- K. J. Cios and L. A. Kurgan. Trends in data mining and knowledge discovery. *Advanced Techniques in Knowledge Discovery and Data Mining*, pages 1–26, 2005.
- J. Cohen. *Statistical Power Analysis for the Behavioral Sciences (2nd Edition)*. Routledge Academic, 2 edition, January 1988.
- G. C. Critchfield, K. E. Willard, and D. P. Connely. Probabilistic sensitivity analysis methods for general decision models. *Comput. Biomed. Res.*, 19: 254–265, June 1986. ISSN 0010-4809.
- R. I. Cukier, H. B. Levine, and K. E. Shuler. Nonlinear sensitivity analysis of multiparameter model systems. *The Journal of Physical Chemistry*, 81 (25):2365–2366, 1977.
- A. C. Cullen and H. C. Frey. *Probabilistic techniques in exposure assessment: a handbook for dealing with variability and uncertainty in models and inputs*. Plenum Press, 1999.
- C. O. Daub, R. Steuer, J. Selbig, and S. Kloska. Estimating mutual information using b-spline functions - an improved similarity measure for analysing gene expression data. *BMC Bioinformatics*, 5(118), August 2004.
- Z. Dawy, B. Goebel, J. Hagenauer, C. Andreoli, T. Meitinger, and J. C. Mueller. Gene mapping and marker clustering using shannon’s mutual information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(1):47–56, 2006.
- K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 edition, jun 2001.
- E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
- R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, New Yotk, 1973.
- R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Pattern Classification and Scene Analysis: Pattern Classification. Wiley, 2001.



- N. F. O. Evbuomwan, S. Sivaloganathan, and A. Jebb. A survey of design philosophies, models, methods and systems. *Journal of Engineering Manufacture*, 210:301–320, 1996.
- J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18:99–119, 1996.
- P. J. Flynn and A. K. Jain. On reliable curvatures estimation. *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 110–116, 1989.
- A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008.
- T. Fukushige, T. Ariyoshi, Y. Uda, Y. Okuma, T. Okabe, K. Yamamoto, L. Graening, S. Menzel, and M. Olhofer. Development of real-time aerodynamic design system. *Honda R&D Technical Review*, 24(2):107–116, 2012.
- L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), September 2006.
- M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.
- L. Graening, S. Menzel, M. Hasenjäger, T. Bihrer, M. Olhofer, and B. Sendhoff. Knowledge extraction from aerodynamic design data and its application to 3d turbine blade geometries. *Mathematical Modelling and Algorithms*, 7:329–350, 2008.
- L. Graening, M. Olhofer, and B. Sendhoff. Interaction detection in aerodynamic design data. In *Proceedings of the 10th international conference on Intelligent data engineering and automated learning, IDEAL'09*, pages 160–167, Berlin, Heidelberg, 2009. Springer-Verlag.
- L. Graening, N. Aulig, and M. Olhofer. Towards directed open-ended search by a novelty guided evolution strategy. In R. Schaefer, C. Cotta, J. Koodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 71–80. Springer Berlin Heidelberg, 2010.



## BIBLIOGRAPHY

---

- L. Graening, S. Menzel, T. Ramsay, and B. Sendhoff. Application of sensitivity analysis for an improved representation in evolutionary design optimization. In *IEEE Int. Conf. on Genetic and Evolutionary Computing (ICGEC)*, pages 1–4. IEEE Press, 2012.
- E. Hameiri and I. Shimshoni. Estimating the principal curvatures and the darbox frame from real 3-d range data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(4):626–637, 2003.
- B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15:1059–1068, 2002.
- J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems Series. Elsevier Science & Tech, 2006.
- N. Hansen and S. Kern. Evaluating the cma evolution strategy on multimodal test functions. In X. Yao et al., editors, *Parallel Problem Solving from Nature PPSN VIII*, volume 3242 of *LNCS*, pages 282–291. Springer, 2004.
- N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolutionary strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0262082136.
- W. M. Hsu, J. F. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 177–184, New York, NY, USA, 1992. ACM Press.
- H. W. Hu, Y. L. Chen, and K. Tang. A dynamic discretization approach for constructing decision trees with a continuous label. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1505–1514, 2009.
- W. H. Hucho. Aerodynamics of road vehicles - a challenge for computational fluid dynamics. In K. W. Seibert and R. K. Hanna, editors, *Proceedings of the 1st European Automotive CFD Conference*, pages 1–16, Bingen, Germany, June 2003. Fluent.
- J. Jaccard and R. Turrisi. *Interaction Effects in Multiple Regression*. Sage Publications, 2003. Second Edition.



- A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.
- A. Jakulin and I. Bratko. Analyzing attribute dependencies. In N. Lavrac, D. Gamberger, H. Blockeel, and L. Todorovski, editors, *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)*, volume 2838 of *Lecture Notes in Artificial Intelligence*, pages 229–240, Cavtat-Dubrovnik, Croatia, 2003. Springer.
- A. Jakulin and I. Bratko. Quantifying and visualizing attribute interactions: An approach based on entropy. <http://arxiv.org/abs/cs.AI/0308002> v3, 308002, 2004a.
- A. Jakulin and I. Bratko. Testing the significance of attribute interactions. In R. Greiner and D. Schuurmans, editors, *Proceedings of the Twenty-first International Conference on Machine Learning (ICML-2004)*, pages 409–416, Banff, Canada, 2004b.
- A. Jakulin, I. Bratko, D. Smrke, and J. Demšar B. Zupan. Attribute interactions in medical data analysis. In M. Dojat, E. Keravnou, and P. Barahona, editors, *Artificial Intelligence in Medicine*, volume 2780 of *Lecture Notes in Computer Science*, pages 229–238. Springer Berlin Heidelberg, 2003.
- S. Jeong, K. Chiba, and S. Obayashi. Data mining for aerodynamic design space. *Aerospace Computing, Information and Communication*, 2(11):452–469, 2005.
- Y. Jin. *Knowledge Incorporation in Evolutionary Computation*. Springer, 2005.
- Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011.
- T. Jost. *Fast geometric matching for shape registration*. PhD thesis, Universit de Neuchtel, 2002.
- M. Kantardzic. *Data Mining - Concepts, Models, Methods, and Algorithms*. IEEE Press, 2001.
- A. J. Keane and P. B. Nair. *Computational Approach for Aerospace Design*. Wiley, 2005.



## BIBLIOGRAPHY

---

- J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.
- S. Kern, S. D. Müller, N. Hansen, D. Büche, J. Očenášek, and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms— a comparative review. *Natural Computing*, 3:77–112, 2004.
- S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.
- T. Kohonen. *Self-Organizing Maps*. Springer, Heidelberg, 1995a.
- T. Kohonen and T. Honkela. Kohonen network. *Scholarpedia*, 2(1):1568, 2007.
- T. K. Kohonen. Learning vector quantization. In M.A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 537–540. MIT Press, Cambridge, MA, 1995b.
- T. K. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–1384, August 2002.
- K. Krippendorff. *Information Theory: Structural Models for Qualitative Data*, volume 07-062 of *Sage University Paper Series on Quantitative Applications in the Social Sciences*. CA: Sage, Newbury Park, 1986.
- K. Krippendorff. Ross ashby’s information theory: a bit of history, some solutions to problems, and what we face today. *International Journal of General Systems*, 38(2):189–212, 2009a.
- K. Krippendorff. Information of interactions in complex systems. *International Journal of General Systems*, 38(6):669–680, 2009b.
- S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86, 1951.
- P. Laskov and R. Kambhamettu. Comparison of 3d algorithms for non-rigid motion and correspondence estimation. In *In Proceedings of the British Machine Vision Conference*, pages 273–282, 2001.



- L. Leydesdorff. Interaction information: linear and nonlinear interpretations. *International Journal of General Systems*, 38(6):681–685, 2009.
- N. Lüdtke, S. Panzeri, M. Brown, D. S. Broomhead, J. Knowles, M. A. Montemurro, and D. Kell. Information-theoretic sensitivity analysis: a general method for credit assignment in complex networks. *Journal of The Royal Society Interface*, 5(19):223–235, 2008.
- D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- J. Mao and A. K. Jain. A self-organizing network for hyperellipsoidal clustering (hec). *IEEE Transactions on Neural Networks*, 7(1):16–29, 1996.
- W. J. McGill. Multivariate information transmission. *Psychometrika*, 19(2):97–116, June 1954.
- F. R. Menter. Zonal two-equation  $\kappa$ - $\omega$  turbulence models for aerodynamic flows. *AIAA Journal*, page 2906, 1993.
- J. S. Mill. *A System of Logic*. University Press of Pacific, 1882.
- G. A. Miller. Note on the bias of information estimates. *Information Theory in Psychology: Problems and Methods*, pages 95–100, 1955.
- R. J. Miller and Y. Yang. Association rules over interval data. In *SIGMOD Conference '97*, pages 452–461, 1997.
- M. T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997. URL <http://www.cs.cmu.edu/~tom/mlbook.html>.
- A. A. Montaña, C. A. Coello Coello, and E. Mezura-Montes. Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Trans. Evolutionary Computation*, 16(5):662–694, 2012.
- M. D. Morris and T. J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43:381–402, 1995.
- R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments (Wiley Series in Probability and Statistics)*. Wiley, 3 edition, January 2009.



## BIBLIOGRAPHY

---

- S. Obayashi and D. Sasaki. Visualization and data mining of pareto solutions using self-organizing map. In *Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization*, EMO'03, pages 796–809, Berlin, Heidelberg, 2003. Springer-Verlag.
- S. Obayashi, S. Jeong, and K. Chiba. Multiobjective design exploration for aerodynamic configurations. pages 1–25, June 2005.
- S. Obayashi, S. Jeong, K. Shimoyama, K. Chiba, and H. Morino. Multi-objective design exploration and its applications. *International Journal of Aeronautical and Space Science*, 11(4):247–265, 2010.
- M. Olhofer, Y. Jin, and B. Sendhoff. Adaptive encoding for aerodynamic shape optimization using evolutionary strategies. In *Congress on Evolutionary Computation (CEC)*, pages 576–583, Seoul Korea, May 2001. IEEE Press.
- C. Othmer. Cfd topology and shape optimization with adjoint methods. *VDI-Berichte, 13th Internationaler Kongress, Berechnung und Simulation im Fahrzeugbau*, 1967:61–72, 2006.
- C. Othmer, E. M. Papoutsis-Kiachagias, and K. Haliskos. Cfd optimization via sensitivity-based shape morphing. In *4th ANSA & mETA International Conference*, Thessaloniki, Greece, 2011.
- G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote. *Engineering Design, A Systematic Approach*. Springer, third edition edition, 2007.
- C. Pateritsas, S. Modes, and A. Stafylopatis. Extracting rules from trained self-organizing maps. In *Proceedings of the International Conference Applied Computing (Salamanca, Spain, 18-20 February 2007)*, pages 183–190, 2007.
- Z. Pawlak. Rough sets. *International Journal of Information & Computer Sciences*, 11:341–356, 1982.
- Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer Dordrecht, 1991.
- D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734, San Francisco, 2000. Morgan Kaufmann.



- C. Priester, K. Narukawa, and T. Rodemann. A comparison of different algorithms for the calculation of dominated hypervolumes. In *Genetic and Evolutionary Computation Conference (GECCO)*, Amsterdam, Netherlands, July 2013. ACM. To appear.
- J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1 edition, October 1992.
- P. Ralph and Y. Wand. *A proposal for a formal definition of the design concept*, volume 14, pages 103–136. Springer-Verlag, 2009.
- M. Rath and L. Graening. Modeling design and flow feature interactions for automotive synthesis. In *Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL)*. Springer, 2011.
- I. Rechenberg. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. PhD thesis, TU Berlin, 1971.
- E. Reehuis, J. Kruisselbrink, M. Olhofer, L. Graening, B. Sendhoff, and T. Baeck. Model guided evolution strategies for dynamically balancing exploration and exploitation. In *The Biannual International Conference on Artificial Evolution*, pages 306–317, Angers, France, 2011. Université d'Angers.
- G. L. Rocca. Knowledge based engineering: Between ai and cad. review of a language based technology to support engineering design. *Advanced Engineering Informatics*, 26(2):159–179, 2012.
- J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, Feb. 1988.
- L. Rokach and M. Oded. *The Data Mining and Knowledge Discovery Handbook*, chapter 15, pages 321–352. Springer, 2005.
- Y. Rubner, C. Tomasi, and L. J. Guibas. The earth movers distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, pages 486–493. IEEE Computer Society, 2004.
- S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *Systems, Man and Cybernetics, IEEE Transactions on*, 21(3):660–674, 1991.



## BIBLIOGRAPHY

---

- A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, and M. Saisana. *Global Sensitivity Analysis. The Primer*, John Wiley & Sons, 2007.
- R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1590–1602, 2011.
- K. M. Saridakis and A. J. Dentsoras. Soft computing in engineering design - a review. *Advanced Engineering Informatics*, 22(2):202–221, April 2008.
- R. Schneiders, R. Schindler, and F. Weiler. Octree-based generation of hexahedral element meshes. In *Proceedings of the 5th International Meshing Roundtable*, pages 205–215, 1996.
- D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, 1 edition, sep 1992.
- T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20:151–160, August 1986. ISSN 0097-8930.
- S. Z. Selim and M. A. Ismail. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):81–87, 1984.
- C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, July 1948.
- S. Shirdhonkar and D. W. Jacobs. Approximate earth mover's distance in linear time. In *Proceedings of the IEEE Computer Society Conf. on Comput. Vision and Pattern Recognition*, pages 1–8, 2008.
- C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, 1998.
- I. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling & Computational Experiment*, 1:407–414, 1993.
- E. H. Spanier. *Algebraic Topology*. MMcGraw-Hill, New York, 1966.
- R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *SIGMOD Conference '96*, pages 1–12, 1996.



- P. Tan and V. Kumar. Interestingness measures for association patterns: A perspective. In *KDD 2000 Workshop on Postprocessing in Machine Learning and Data Mining*, Boston, MA, 2000.
- P. Tan, V. Kumar, and J. Srivastava. Selecting the right objective measure for association analysis. *Inf. Syst.*, 29(4):293–313, June 2004.
- G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *In proceedings of the 5th International Conference on Computer Vision*, pages 902–907, 1995.
- R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B*, 63(2):411–423, 2001.
- A. Ultsch. Knowledge extraction from self-organizing neural networks. *Information and Classification*, pages 301–306, 1993.
- A. Verma. An introduction to automatic differentiation. *Current Science*, 78(7):804–807, 2000.
- Y. Wang, B. S. Peterson, and L. H. Staib. Shape-based 3d surface correspondence using geodesic and local geometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 644–651, 2000.
- G. I. Webb. Discovering significant rules. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 434–443, New York, NY, USA, 2006. ACM.
- G. I. Webb and S. Zhang. Removing trivial associations in association rule discovery. In *Proceedings of the First International NAISO Congress on Autonomous Intelligent Systems (ICAIS 2002)*, Canada/The Netherlands, 2002. NAISO Academic Press.
- L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *Evolutionary Computation, IEEE Transactions on*, 16(1):86–95, 2012.
- L. A. Zadeh, G. J. Klir, and B. Yuan. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*. Advances in Fuzzy Systems. World Scientific, 1996.
- G. P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, November 2000.





