Exploring Automotive Shape with Kansei Design

— A Systematic Approach to Building Design Support Systems with Shape Sensibility —

September 2007

NORDGREN, Andreas Kjell

Acknowledgements

I would like to thank the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) for providing me with the Monbukagakusho scholarship during my years at Keio University, without this financial support I could not have undertaken this study. Professor Hideki Aoyama has also been helpful in supervising my efforts, and allowing me to delve into different fields of research, so that I could draw inspiration and find my own methodology in Kansei Design. Finally, I thank professors Masafumi Hagiwara, Yoshiyuki Matsuoka and Hiromitsu Ohmori for their insightful comments on my work.

Abstract

In this thesis I will present a methodology for exploratory analysis of emotional content in shapes. My aim is to provide a broad overview of some of the tools needed, as well as to lay out a systematic approach to system implementation, which includes automated construction of design concepts via Design of Experiments (DOE), data-collection via on-line survey systems, data analysis with Principal Component Analysis (PCA), model-building with Artificial Neural Networks (ANN), and shape construction and parameterization performed both by an explicit and implicit approach.

The range of technologies used is therefore by necessity broad, but I still hope that the casual reader can get an idea of the work-flow and general idea behind Kansei studies, while still providing enough details for system developers to integrate these methods into CAD-programs that can support design decision based on subtle consumer preferences in the emotional domain.

Contents

List of Tables

List of Figures

Nomenclature

Roman Symbols

$\mathbb{L}^2$ Hilbert space

$R(x, x')$ matrix of covariances between scalar components of vector $\bar{X}$

r radius of repulsion; used in algorithm to create uniformly distributed point-cloud on polygonal mesh

$u(x, t)$ ensemble of design concepts, where t is the index of each concept (point-cloud)

$v(x, t)$ deviation from mean (dataset of point-clouds); a measure of unique shape characteristics

Greek Symbols

λ eigenvalue of process $v(x, t)$

φ eigenvector of process $v(x, t)$

Acronyms

ANN Artificial Neural Network

CAD Computer-Aided Design

TTM Time-to-market

CAE Computer-Aided Engineering

CAM Computer-Aided Manufacturing

CFD Computational Fluid Dynamics

DOE Design of Experiments

$EvokeDB$ Evaluation and Verification of Objects in Kansei Engineering via Database

KE Kansei Engineering

$OFAT$ One-factor-at-a-time

PCA Principal Component Analysis

PLM Product Lifecycle Management

Chapter 1

Introduction

The industrialized world has seen a vast range of products offered to make our lives more comfortable – some brought completely new features which became necessities, and thus launched products into top-sellers, at least until the competition caught up and brought something better or cheaper.

The history of product development is well-documented with the icons of each century, and whilst there have been many new advances and success-stories, this path has also been littered with the forgotten shells of products that failed to attracted customer attention.

So what makes a product successful? Well, if I had the answer to that question, I would probably be in a more profitable line of business by now. Every company competing in an open market has to focus on this issue, and they do so by analysing the market, improving their products and manufacturing methods so that hopefully they can get an edge over the competitors and appeal to a group of people like you and I, because we ultimately buy the products we have a preference for. Knowing or forecasting consumer preferences, and creating innovative products by design, thus seem to be key ingredients to success in this game.

Marketing, innovative or relentless, is of course also a very important tool to create the brand image and lifestyle associated with the product. These very subtle signals of a product are becoming increasingly important in a competitive market where product differentiation among makers' is often on a level where the customer's basic needs and wants are already covered [43].

Consider the automotive industry and the process of buying a new car. This is a major purchase for most people, and therefore the various requirements, that is, the “needs” and “wants”, have been considered at some level. Based on these needs and wants, the customer is directed to a specific segment of the market. However, due to the competitive nature of the automotive industry, within any segment you can walk into a dealership and buy a car that fulfills your basic requirements on safety, performance, fuel-economy and so on. The purchase-decision is, therefore, also highly influenced by emotion. This means that how you “feel” about the car and its attributes on some internal level can be a deciding factor in your purchase. This is a simplification, but it still holds true that the automotive industry is highly competitive and customers can choose between very similar models offered by different car makers, so that technology in itself is no longer a deciding factor in the purchase-decision [33].

So how can feelings matter more than reason in this type of situation? Take safety features, which are considered as important by most people – these are usually described and presented in catalogs, brochures or other media, and may in fact not be clearly visible in the car itself. This means that many people will treat these as abstractions that can be logically understood, although they offer no means for us to explore them by our senses. Everybody understands that an airbag is good, but we cannot have any feedback or judge its quality by physical senses unless we actually have an accident. The form of a car can, however, carry features that customers can explore and form an intuitive feeling about, and these features can be used as sensory cues to express the concept of “safety” more strongly than any logical explanations can communicate. A study by Creusen and Schoorman [8] identified six different roles of product appearance for consumers as:

1. Communication of aesthetic, and

2. Symbolic, and

3. Functional, and

4. Ergonomic information, and

5. Attention drawing, and
6. Categorization.

A product's appearance should therefore not only be seen from the viewpoint of aesthetics, but as a carrier of symbolic values, functional characteristics, quality impressions, usability and ergonomic values. It can also be used to place the product in a specific category, and further communicate the associated lifestyle.

However, this type of added content in products is far more difficult to measure and understand than traditional methods based on specifications and functions of technology that are clearly defined and measurable; in this case we must try to understand very subtle impressions evoked from a product, where a customer explores the product with all senses, makes associations to past experiences and desires to form an emotional basis that plays a very important part in the decision-making process.

In this thesis I will present a methodology for exploratory analysis of emotional content in shapes. My aim is to provide a broad overview of some of the tools needed, as well as to lay out a systematic approach to system implementation, which includes construction of design concepts based on Design of Experiments (DOE), data-collection via on-line survey systems, data analysis with Principal Component Analysis (PCA), model-building with Artificial Neural Networks (ANN), and shape construction and parameterization performed both by an explicit and implicit approach. The range of technologies used are therefore by necessity broad, but I still hope that the casual reader can get an idea of the workflow and general idea behind Kansei studies, while still providing enough details for system developers to integrate these methods into CAD-programs that can support design decision based on subtle consumer preferences in the emotional domain.

1.1 What is ***Kansei***?

I have chosen to include the term "Kansei" in the title of my thesis as I wish to promote its use in design over descriptors like "Design for Emotion", "Affective Design" and "Emotional Design". Kansei has a rather broad and fuzzy meaning, depending on the discipline and its practitioner, and as an engineer I initially found it hard to cope with the lack of a clearly defined terms.

The word “Kansei” has been used in philosophy, epistemology, psychology and now engineering, with some variations in the definitions, but generally describing how a human subject can sense external events by sensory organs, which triggers an internal emotional response that is used for judgement.

Kansei is often seen in contrast to reason and intellectual abilities based on facts (e.g. *Chisei*), but it is still widely accepted that Kansei, as a sensuous intuition, plays an important role in how we as humans understand, evaluate and interact with objects in our environment.

According to Shinya Nagasawa [32], a major Japanese dictionary gives a definition of Kansei as:

> Sensibility of a sensory organ where sensation or perception takes place in answer to stimuli from the external world.

A simple descriptor could then be that Kansei stands for “sensitivity”, although this does not include all the facets of Kansei associated with the kanji-characters used in Japanese documents. Nagasawa also gave a good definition of Kansei [32], with three levels of simplification, as:

> *Kansei*, if used in engineering and business, should be considered to be a series of information processing processes of sensation, perception, cognition, sentiment, and expression on the basis of the definition of *Kansei* of information-processing cognitive psychology, or it would be enough, more practically, that *Kansei* should be though to be a series of reaction from sensation to mental responses, or much more simply, sensation and sentiment.

A study by Harada [18] showed that there are many interpretations of Kansei among the researchers in Japan. He found five main aspects of Kansei, constructed from clusters of definitions gathered from a group of researchers, as:

1. Kansei is a subjective effect which cannot be described by words alone.

2. Kansei is a cognitive concept, influenced by a person's knowledge, experience, and character.
3. Kansei is a mutual interaction between the intuition and intellectual activity.
4. Kansei entails a sensitivity to aspects such as beauty or pleasure.
5. Kansei is an effect for creating the images often accompanied by the human mind.

These definitions can guide the budding Kansei practitioner, but I believe that the precise definition of Kansei in engineering is not crucial, and that the fuzziness of these descriptors is unlikely to cause much concern for most product developers.

1.2 History and origin of Kansei Design and Engineering

Kansei, although recently popularized world-wide as a tool in product development, has been an active and growing research field in Japan since the 70's, when Professor Mitsuo Nagamachi (then of Hiroshima University) started working with methods to consider emotions in product development. He is widely recognized as a forerunner and active practitioner of Kansei Engineering, both in Japan and world-wide, with many published papers and a book [31] in English. There is a lot of knowledge about Kansei Engineering in Japan, some of which have been presented as an overview [17], but the reality remains that many papers are only published in Japanese, although the last 10 years has seen a substantial growth of international papers.

The very idea of Kansei is not a new phenomenon, it has of course been used ever since the first caveman chose a particularly robust and ergonomic club, but Kansei Engineering as a set of methods was formalized from the 70's and onwards. The 80's brought a lot of attention to Kansei in the marketing industry in Japan – terms such as *Kansei consumption* and *Kansei marketing* became buzz-words that indicated that products were no longer judged by reason alone, but on the basis of subtle emotional likes and dislikes that elicited specific customer preferences.

However, trends in marketing are usually short-lived, and Kansei Engineering has now started to mature and attract attention outside of Japan. There are now many research groups based in European universities, such as Delft University in Holland, Linköping University in Sweden, and Leeds University in the UK. Several companies have also started to build up their Kansei Engineering competencies, as evident by the increasing number of industry participants in the Kansei related conferences.

So why is Kansei becoming more important today? One reason is that we have entered the digital era where our high-tech lifestyles are filled with gadgets, and many of our interactions take place on-line, or with machines. Kansei is therefore becoming a very important tool to bring more life-like features to products and services that have in many cases replaced face-to-face interactions between humans. There are many different disciplines where Kansei studies are actively used to improve the emotional aspect of objects and environments in our daily life. This becomes apparent at the conferences in Kansei Engineering that are attracting an increasingly diverse crowd every year – product design, health-care, e-commerce, creativity, services, signs and almost every conceivable system that involves human emotion is actively investigated.

Kansei Engineering still feels like a very new research field where new methodologies and applications are introduced every year, and as a novice it is quite difficult to grasp the shear number of statistical tools and algorithms used. It is a very “soft” science, and by default there are many different entry-points and approaches to extract emotional meaning from the artifacts around us. This “fuzziness”, coupled with the multidisciplinary nature of the field, does bring some extra challenges for product developers. However, I believe that we will see a more formalized nomenclature and a higher level of usability in Kansei Engineering the coming years, as results from the academic domain find the way to applications in the industry.

1.3 Kansei and cars

Cars carry emotional value like few other objects. It can symbolize freedom and status, and most people have a very clear attitude about cars, both positive and negative. One thing that is for sure is that purchasing a car is usually a major decision since that car will be a very important object for years to come.

We can identify and express ourselves, or the person we want to be, by the car we choose to drive. Granted, for some people the basic functionality as a mode of transportation from A to B is most important, but it also seems to be the case that the relationship with the car goes much deeper than that. As consumers we are rather spoilt and can choose from a wide range of brands that will cater to our individualistic needs and wants. The competitive nature of the automotive industry has led to highly developed products that cover our basic requirements of quality, safety, performance and so on. Car makers must therefore differentiate themselves and build the brand image of their cars to attract a certain segment of the market. This has led to cars being the most developed consumer product on the market today; companies are spending huge amounts on product development, and you might be surprised to learn that even the sound of a closing door, the tactile feel of buttons and the tread-pattern on the floor carpets have been put through extensive studies to optimize the design to meet your desires (or perhaps more accurately, the desires of the particular demographic you belong to). Understanding Kansei, and communicating the right Kansei is therefore essential for product success in the automotive industry.

All types of parts in a car are important for how we as customers perceive it, but most important of all is probably the shape of the car itself. This is the first impression we get, by a visual inspection in a showroom or as a quick glance at the street as a new model passes by. The shape, either seen as the global form, or in combination with very small details such as shut-lines, characteristic curves and proportions, can carry tremendous meaning to us – we will get an intuitive feeling for the car's robustness, safety, performance and so on simply by looking at its shape.

1.4 Previous work

My work involves two main areas – Computer Aided Design (CAD) and Kansei Engineering (KE), and the previous work related to my thesis is therefore very broad. CAD has become an indispensable tool in product design, development and manufacturing, and the entertainment industry, primarily computer games and computer graphics (CG) in movies, has also spurred on the development of new and improved tools to create and visualize artifacts in 3D. However, most of these tools aim to give a more efficient representation of surfaces, or allow more intuitive deformations such as cutting, merging, morphing or transforming 3D-models to create new concepts or automated animations.

Shape morphing, or interpolation, has been used by Chen et al. [7] to explore emotional content in automotive shape spaces. However, this approach is fundamentally different from the methodology I present in this thesis. Interpolation does not model any underlying relationships between shapes and the emotional response they evoke; instead, data is collected on a number of artifacts, and desired shapes are generated by interpolating artifacts with desired qualities. This is an implicit approach, which does not guarantee that the final shape is accurate, since a linear process has been used to map a problem that very well might exhibit non-linear qualities.

The Kansei Design Methodology in this paper uses statistical methods to obtain equations relating product attributes to perceptual qualities, and this is therefore an inherently more complex approach.

There has been several publications on the methods used in Kansei Engineering (notably [10, 11, 17, 27, 30, 31]) but these outline a general framework. My aim for this study is to provide a systematic approach for Kansei methodologies applied to shape design.

Parts of this work does overlap with well-known methods in KE, but I also provide new insights into the particulars of the implementation of more sensible CAD-systems, where these methods can support a designer and shed some light on the intricate relationships between shape and emotions.

Furthermore, I outline an originally developed system for data-collection in

Kansei studies, and provide some guidelines for using representations of real artifacts.

Finally, I present a novel method for implicit definition of shape parameters based on Principal Component Analysis of an ensemble of surfaces (point-clouds). This gives a very low-dimensional descriptor for complex surfaces that previously have been challenging to analyze due the the large number of parameters needed.

1.5 Structure of thesis

I have divided this thesis into four main chapters:

Chapter 2 contains the Kansei Design Methodology I use as a basis for my work. This involves several statistical methods, notably Principal Component Analysis (PCA), Design of Experiments (DOE) and Artificial Neural Networks (ANN), and their use in this methodology. I will not give any in-depth description of the methods themselves as they are well-documented in the literature.

Chapter 3 describes my ideas on Kansei Fidelity, that is, how representations of real artifacts can fail to evoke Kansei. A pilot-study comparing data measured with an image and a physical scale-model is presented, along with some guidelines for using representations.

Chapter 4 outlines EvokeDB, a survey management system that can be used to automate many tasks in launching on-line surveys with a database back-end.

Chapter 5 describes the foundation and use of Implicit Shape Parameterization: a novel approach to shape space definition based on 3D-modeling.

Each chapter has an introduction in the beginning and a summery in the end, to allow casual readers to grasp the main ideas and read each chapter separately.

Chapter 2

A Kansei Design Methodology

Kansei Engineering and Design has become a popular research topic recently; the number of international conferences on Kansei is increasing, and there are many active research groups, not only in Japan, working on a wide array of problems ranging from fundamental methods all the way to real applications in the industry.

However, there is still a lack of reference-literature available although the methods are increasingly becoming recognized as a powerful set of tools in the product development process. The implementation of systems, and application of Kansei methodologies to solve real problems are difficult to grasp for novice or non-experts. Most studies are built upon methods that are well-documented in their own domain; factor analysis, neural networks, genetic algorithms, rough sets, design of experiments are just some of the procedures that have been used successfully for many years in various research fields. This wealth of tools is very valuable to the research community, but a product developer will no doubt find it difficult to understand what approach to take to a specific problem. What is the work-flow, what are the steps needed, which tools are the best match for my problem? These are questions that need some guidelines, and this chapter presents a systematic approach to at least one very common problem relating to shape design.

My discussions are focused on parts in the automotive domain, but it should at all times be clear that this methodology can easily be generalized to any product with a shape that can be parameterized and described numerically. This methodology is also closely coupled to CAD-systems, and it is my intent and hope that most steps can be implemented in a CAD-tool in such a way so that the underlying complexity of the data analysis can be kept in the background.

CAD is a crucial ingredient in modern product development. Fierce competition in the automotive industry has led to shorter time-to-market (TTM), where auto makers must be very flexible in order to respond to changes in the market. It is absolutely essential to have access to modern CAD/CAM-systems to develop new cars in a short amount of time. A new model needs not only to fulfill the often conflicting demands of the potential customers, but it must also bring something new to capture the interest of the market. Improved technology can bring new functions, features and outstanding quality, but innovative design can be used to create a very important product differentiation.

Design ideas have traditionally been explored through sketches or 3D-renderings, where focus groups can offer feedback on various design features. Therefore, there are methods to include the customer's voice in various stages of the development of a new model, but these do not provide any mathematical model to map form impressions to design. Although useful, focus groups only provide a designer with an intuitive feel of how to meet their desires; he or she must still interpret the subtleties of form impressions, based on limited data. Whereas customer demands for properties such as passenger space, fuel economy, performance and safety are easy to quantify and consider in the design process, the form impression is a subtle association or a feeling, and thus far less easy to relate to design parameters. Such a form impression is often expressed in natural language, which is rather subjective and differs from customer to customer.

CAD/CAM-systems are very helpful in the design and manufacturing process, but they have no sense of creativity or knowledge about the Kansei evoked by shapes. It has always been the role of the designer to interpret subtleties in customer demands for a design, that is, to transfer desires into surfaces and curves, and create a product that will appeal to the targeted group of customers. However, this is not an easy thing to do – it requires experience and sensitivity to current trends, or better yet, the trend for the coming years.

For instance, what shapes and curves will give an impression of a sporty and powerful car, yet with a classic look and elements of formality in it, to a young woman? Is there some set of proportions that will yield a form impression that coincides with what the targeted customer desires? Knowing such a set of proportions would be very valuable in the creation and development of successful cars. A good designer probably has a feeling for these proportions, but it would certainly be valuable to also have this type of sensitivity built into a CAD-system, and be able to visualize how various form impressions will affect the design parameters.

By learning to map a semantic Kansei-space to shape parameters, this type of system could be used in the early stages of the design process to explore shapes, or by the reverse mapping, validate how the form impression changes after a re-design. The methodology presented in this chapter can be used as a foundation to develop a design support system with shape sensibility, and I also hope it will give clear guidelines to any product developer wishing to apply Kansei design principles to his or her next project.

2.1 Design Study Work-flow

I have devised the work-flow shown in Figure 2.1 as an approach for exploratory Kansei studies on shape. Each step involves additional work, but this overview gives an idea about how to structure and perform the necessary tasks in a systematic manner.

2.1.1 The Kansei space – selecting meaningful words

Chapter 1 gave a few definitions of Kansei, and for studies on shape we can consider it as the customer's internal emotional response to the shape we present. By this definition it is natural that Kansei data is not something that is readily available or even measurable without considerable effort. It is a very subtle signal, which by its intrinsic properties is difficult to understand, and even more challenging to put a numeric value on.

There are several entry points that we may consider, but most research up until today have focused on two different approaches. One is measurements of physiological signals from a subject – these are collected as EEG, Galvanic skin response, brain-waves and so on to measure the strength of a response, although

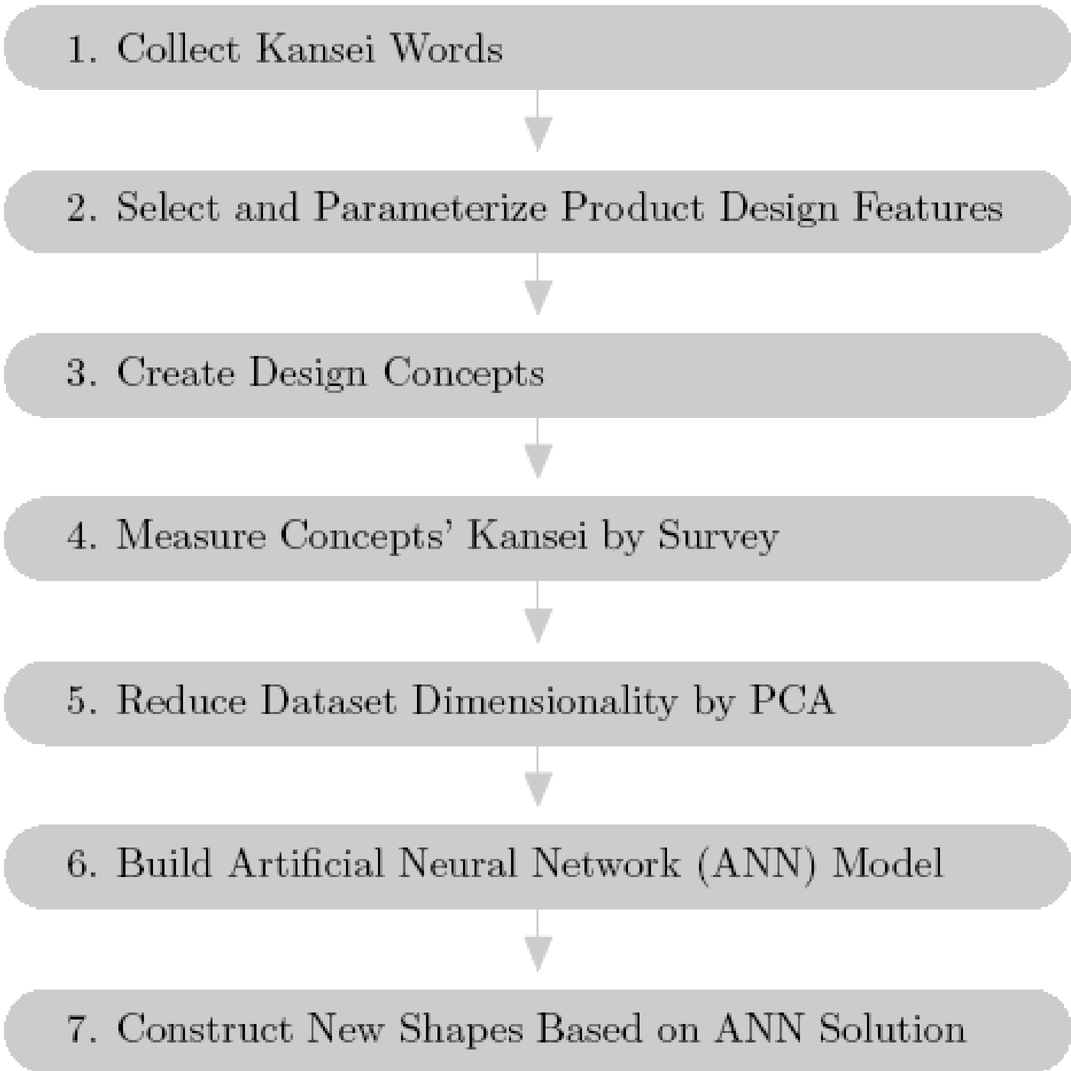

Figure 2.1: Work-flow in Kansei design study of shape

it may be difficult to extract the emotional content of the signal unless other techniques are used in combination. The second approach, which I have adhered to, is to use a psychological measure based on Kansei words. This is where subjects can externalize and evaluate their perceived feelings about a product or shape by placing it in a semantic space spanned by attributes that can describe feelings. This is of course a simplification of a true internal emotion, but one that is necessary in order to carry out quantitative studies with a large number of subjects. The set of Kansei words should be associated with the product in question. There are no rules set in stone when it comes to selecting suitable words, but I will suggest two separate approaches depending on the type of survey being performed.

Let's first consider an exploratory analysis of the product domain. In this case we are interested in finding all the words or expressions that are in some way associated to the type of product we have. Collect words from magazines, advertisements, product manuals, company websites, interviews with users, or any other source linked to the product or associated lifestyle. This is not restricted to just words such as adjectives, it could also be expressions like "zoom-zoom" or "fun to drive", which are related to brand images. Cluster analysis can be used to find words relating to the domain from very large collections, so this is also an efficient method to add words to the set. A set generated in this manner will not be extremely focused, but it can be very useful in some cases when we want to understand the dimensionality of the domain, and where our product and the ones from the competition are located.

The second approach is more useful for specific studies focused on the particular market segment our product should attract. Words here will relate to the company brand image, the demographic we wish to appeal to, and the desired image of our product. The selection of Kansei words is very important, but this area is fuzzy by nature. In a sense, Kansei Design should be seen as a navigation-system – it can guide you along the way, but you must tell it where you want to go first. Most projects will have a very clearly stated group of potential customers based on market research, and this can give a good indication of the set of Kansei words that should be used.

Words with a similar meaning can be used, even though this is not the most efficient way to carry out surveys (this overlap of meaning will be removed by PCA in a later stage). However, it is important to avoid words that can have different meaning among the group of subjects, that is, some words have completely different meaning depending on context and user. A trivial example would be slang-like word like “cool”, “fat” or “hot”, these can have very different meaning to different people, and should therefore be avoided, unless the discrepancies are properly investigated and included in the model. The distribution of responses from subjects can reveal words with duality in meaning – double, or several peaks in the distribution of replies indicate that subjects do not agree on the meaning of the word. Observations with a Gaussian distribution (bell-shape) give more confidence in that the survey is actually measuring the same Kansei for the whole group of subjects.

Table 2.1 shows a set of Kansei words, and this set was used in the study that I present as an example in this chapter. A set of ten attributes were chosen as parameters to represent a form impression. As this study focused on desired features of a car, Kansei words with positive connotations were selected. A 5-point scale was used to put weights on each word, thereby yielding a Kansei vector **F** that describes the form impression of a shape, as perceived by a subject. A higher weight signifies a higher correlation to the word.

Table 2.1: Weighted set of Kansei words describing a form impression

Cute	**Sporty**	**Classic**	**Formal**	**Powerful**
2	5	3	3	4
Modern	**Robust**	**Spacious**	**Sleek**	**Luxurious**
4	4	1	4	3

2.1.2 Parameterize design features – shape descriptor

The second step of the methodology is to decide a set of design parameters that can adequately describe the shape we are working with. For this work, a sedan type of car was considered and 12 parameters for the basic proportions of the side profile shape were decided, as shown in Figure 2.2. Parameters defined as angles and lengths provided a basic control polygon, which later (see §2.5) was used to place the curves and surfaces for the car's body.

Different types of vehicles have some fundamental differences in their basic shapes, and therefore the design parameters for one type of car do not necessarily correspond well to that of another type. Furthermore, the decision of design parameters is not limited to basic proportions of the car. The parameterization of design features is ultimately decided by the type of shape and the feature we wish to investigate. This means that some shapes are best represented by characteristic proportions or control-points of a curve, while others may require a more complex approach (see Chapter 5). Depending on the type of shape on which the designer is working, it is possible to define design parameters freely, as long as they have a significant effect on the shape. In most cases the designer has many constraints to follow, and these constraints should be carefully considered when deciding the design parameters and their ranges (see §2.1.2).

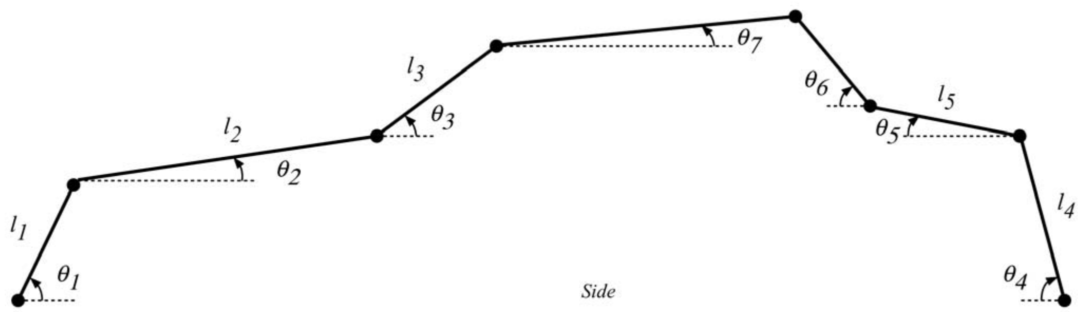

Figure 2.2: 12 design parameters for side profile shape

Only two parameters were used for the cross section shape (seen from the front) of the car in order to minimize the samples needed for the survey. The focus was on proving that the method works rather than to make a production-quality model, and therefore this simplification was made. The parameter α_1 sets the tumblehome, which is the angle of glass from the shoulder-line to the roof as viewed from the front or the rear of the vehicle. The shoulder-width of the car is controlled by w_1.

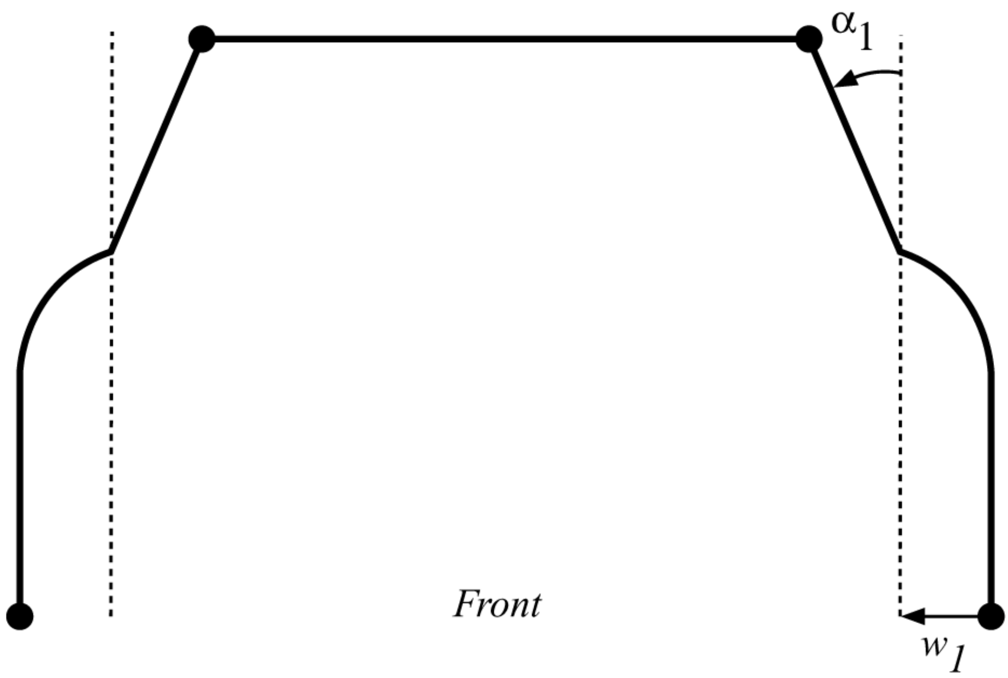

Figure 2.3: Two design parameters for cross-section profile shape

2.2 Design of experiments (DOE)

Design of experiments (DOE) is a systematic approach to investigate a formulated problem, which could be a system or process with clearly defined and measurable inputs and outputs. A series of structured tests is designed, in which planned changes are made to the input variables of the process or system, so that the effects of these changes on a pre-defined output can be analyzed with statistical tools. The methodology presented in this chapter uses DOE to create design concepts, which are subsequently used in a survey to gather data on how subjects relate to various design features of an artifacts' shape. DOE plays an important role here, as a formal way of maximizing information gained while minimizing resources required, which in this case means reduced work to carry out for the survey; we are effectively given a smaller set of design concepts, where the effects of any one factor are balanced out across all other factors.

DOE offers far more insight into a problem than any of the simplistic "one-factor-at-a-time" [9] (OFAT) experimental methods, because it allows a judgement on the significance to the output of input variables acting alone, as well input variables acting in combination with one another. OFAT experiments has the risk that the experimenter may find one input variable to have a significant effect on the response (output) while failing to discover that changing another variable may alter the effect of the first. This is a problem of interaction between input variables, which must be actively explored by the researcher to create an accurate model or the problem. DOE, on the other hand, plans for all possible interactions among the input variables, and then prescribes exactly what data are needed to assess them; will input variables change the response on their own, when combined, or not at all?

For this Kansei design methodology, DOE is used as a sampling algorithm to decide how the shape parameters of §2.1.2 should be combined in a small number of design concepts, but still represent the whole design space in a uniform manner. This implies a balanced set where each shape parameter will be judged by the same number of observations. Orthogonal arrays will be used for this, and the necessary resources can therefore be estimated directly, as the exact length and size of the experiment are set by the design.

As explained in §2.1.1, the order of tasks to using this tool starts with identifying the input variables and the response (output) that we seek to measure. For each shape parameter in the input, a number of levels are defined to represent the range for which the effect of that parameter is desired to be known. This range should be decided based on design constraints of the shape, however for the exploratory analysis of basic proportions presented in §2.2.2 the permissible ranges were exaggerated to yield concepts that were easily separated in a pilot-survey.

DOE gives an experimental plan that tells the experimenter where to set each test parameter for each run of the test. The response is then measured for each run, and collected into a database. DOE offers many different tools to analyze this data [5], and find optimum solutions or robust designs [42] that are insensitive to noise factors. However, the coming sections will describe a different approach where the objective is to map input to output by means of a neural network, which in essence will allow a designer to explore every aspect of how Kansei is evoked by design, and vice-versa. Some background and considerations to the available approaches for the design of experiments are given below.

2.2.1 Orthogonal arrays

A mathematical description of orthogonality would be to consider two vectors of the same length, which are orthogonal to each other if and only if the sum of their corresponding elements is zero. In the field DOE, an orthogonal array carries a meaning of “balanced”, “separable”, or “not mixed”. Dr. Genichi Taguchi has presented a set of orthogonal arrays: so called “L-arrays”, which can assign a number of variables to measure a process or system response in a very economical manner [42]. His work with orthogonal arrays focuses on reproducibility, which means that effects should be reproduced in the following situations:

1. The conclusions from the test piece study are reproduced in the actual product.
2. The conclusions from a small-scale study are reproduced in large-scale manufacturing.
3. The conclusions from limited conditions are reproduced under various other customers’ conditions.

In the case of this methodology, we are mostly interested in finding a subset of all possible combinations of input variables and their levels, such that we can perform a survey efficiently and still have confidence in the conclusions drawn from the results. The shape parameters are assumed to be independent from each other, and any interactions will, by the design of the orthogonal arrays, be almost evenly distributed to other columns and confounded with main effects. Appendix A contains the tables of orthogonal arrays L_9, L_{12}, L_{18} and L_{27}.

2.2.2 Creating concepts by DOE

A survey was designed and conducted in order to collect data on the form impressions of various shapes. The aim for this survey was to prove the methodology by conducting a small pilot-study, and the results should be viewed with consideration to the scarce amount of data collected. An in-depth analysis of form impressions for cars will require more samples and a larger number of subjects.

Every design parameter will give a contribution to the form impression, and therefore all parameters must all be accounted for and changed uniformly for the samples included in the survey. To facilitate the creation of samples and ensure a minimum of samples with uniformly changed parameters, Design of Experiments [42] (DOE) was used to select a sampling of all possible states.

Each parameter for the side profile shape was given three levels to change between. The L_{27}-array (see Appendix A) provided 27 different runs of the experiment, or in this case, samples of design concepts. Thus each sample consisted of the silhouette of the side profile of the car, as displayed in Figure 2.4.

The front parameters were restricted to two levels in order to minimize the contribution to the survey with four samples (omitted from Figure 2.4). Furthermore, the survey was divided into separate parts for the cross section and the side profile, in order to simplify the survey by presenting 2D-silhouettes instead of more complex 3D-models. Chapter 5 describes a method to use complex 3D-models as concepts, and the extension of this methodology to 3D-surfaces is therefore possible.

Via a web-browser interface [35], these samples were presented to 21 individuals consisting of men and women in the ages between 20 and 30 years old. Each individual was given definitions of the attributes from the Merriam-Webster Dictionary™, and a brief discussion also took place to make sure the attributes were clearly understood. This was a very special case since all subjects were of Japanese nationality and therefore only spoke English as a second language. In general, provision of a dictionary entry may introduce biased results, and surveys should always be performed with the subjects mother-tongue. Kansei words depend on associations that may not exist, or have strength, for a second language although the meaning is understood.

The "average" model in Figure 2.9 was created with the parameters given by the second (average) level in the range. It was displayed as an example to each subject before the survey was given. The form impressions for the samples were collected and saved in a database. When the survey was completed by the group, the average response for each model was calculated and used as a value to describe the form impression.

For this work, the permissible ranges of the levels were rather large in order to create slightly exaggerated samples which would be easier to separate for the subjects taking the survey. For example, the hood was given levels of 30-, 50-, and 70-length units to change between. In a production environment, the design team will already have an idea or concept of what type of car they will develop, and they will also have to consider many engineering and manufacturing constraints. These factors will lead to a much smaller range of permissible values for the levels, and thus create more realistic samples which show less variation.

Global design constraints can be very difficult to control with this method. It is easy to define a range for individual parameters, but the combination of them may yield shapes that are very peculiar. However, the results of this system is for exploratory analysis of shapes, and the focus should be on giving a designer an idea of what features are important, and which Kansei they are associated with. Automated shape construction will therefore not give a final shape that is within specifications, but it will provide a starting point for refinements, and serve as a tool for validations, as well as to support creativity by exploring and playing with different combinations and strength of Kansei words.

2.3 Principal Component Analysis

The data from the survey provided a multivariate dataset of 12 variables measured over 27 samples (for the side profile shape), yielding a 27x12 primary data matrix. However, this dataset showed that some attributes had similarities and were correlated to each other. To eliminate this overlap of meaning, Principal Component Analysis (PCA) was used to extract correlated factors. The goal of PCA is to, via analysis of eigenvectors and eigenvalues, find a transformation matrix that will provide a new set of coordinate axes where the data can be projected

Figure 2.4: 27 samples for survey (side profile shape)

in such a way that the variance is maximized along subsequent, orthogonal axes (Principal Component Axes). As each extracted component accounts for less and less variance in the data, it is possible to obtain a reduction of parameters while preserving most of the information in the dataset [15]. This reduction in parameters is important in order to lower the dimensionality of the problem which the neural network must solve – according to Friedman [14], a function defined in high dimensional space is likely to be more complex than one in a lower dimensional space, and therefore harder for the network to solve.

The number of components to extract is rather subjective and there are many techniques available to aid in this decision. Generally, a sufficient number of components must be extracted to accurately reproduce the data matrix from component loadings and component scores, but the goal of this analysis is still to extract a limited number of components that will contain the maximum amount of information.

A Scree Test [6] was performed but provided inconclusive results. According to Figure 2.5, the number of extracted components should be either four or six. The Kaiser Criterion [23], which states that only components whose eigenvalues are greater than 1.0 should be kept, suggested that the first three components would suffice to express most of the information contained in the dataset. Therefore, with the results from these two tests, I decided to retain four components, which accounted for 90.3% of the variance.

In addition to extracting the principal component axes, the PCA also provided the component loading matrix which contains the variable loadings on each of the retained axes, thus displaying the correlations between the attributes and the components. PCA was chosen over Factor Analysis as we were mostly interested in reducing the dataset. However, the collected data can also give some very good knowledge about how different Kansei words relate to each other. In this case a factor analysis with Varimax-rotation will give a good model of these relationships. Varimax-rotated factors [19] will force the factor-axes to align as closely as possible with strongly correlated subsets among the form impression attributes. The idea behind this procedure is to obtain a simpler structure which helps in the interpretation and labeling of the factors. In a simple structure each factor loads highly on a few variables, and each variable loads highly on only one

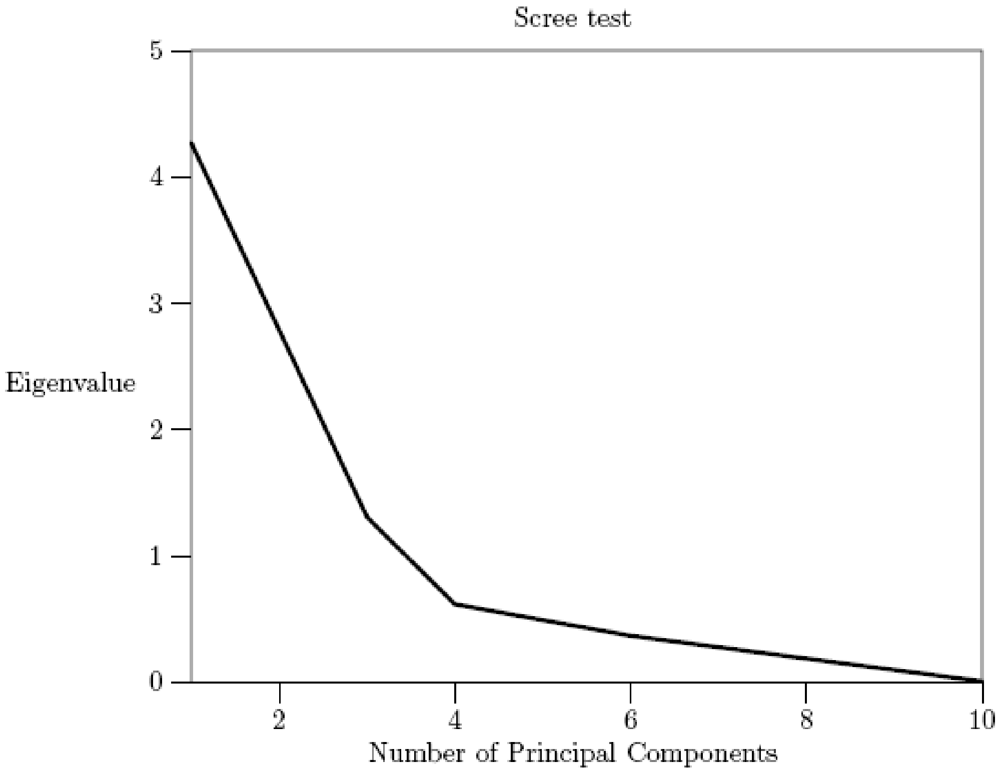

Figure 2.5: Scree test to determine number of retained components

factor. Furthermore, variables that load heavily on the same factor are related, whereas unrelated variables would load on different factors.

For the significance of these loadings, Hair et al. [16] suggested a guideline where loadings larger than ± 0.5 can be considered as practically significant, that is, they have a meaningful effect on the variables. An example of component loadings are displayed in Table 2.2. An analysis of this table can give an interpretation of the components, but for the function of the system described in this paper, a formal labeling was not necessary, and typically PCA for dataset reduction is sufficient, as we are mainly seeking to remove extraneous data and reduce the dimensionality of the problem so that the neural network mapping (see §2.4) becomes easier.

Table 2.2: Principal component loading matrix

Attribute	**PC1**	**PC2**	**PC3**	**PC4**
Cute	-0.3503	-0.0882	-0.0119	-0.9310
Sporty	0.9242	-0.1311	0.1107	0.1958
Classic	-0.2124	-0.1413	-0.9358	-0.0180
Formal	-0.1812	-0.3151	-0.8984	-0.0209
Powerful	0.0667	-0.9140	-0.0122	-0.0154
Modern	0.5645	-0.2499	0.6455	0.0994
Robust	-0.1126	-0.8404	-0.4131	-0.1333
Spacious	-0.8648	-0.3621	-0.1074	-0.1452
Sleek	0.9188	-0.1451	0.2169	0.2035
Luxurious	0.3602	-0.5521	-0.5980	0.1100

With this component loading matrix it was possible to project the samples, expressed with their original form impression attributes, into the new space spanned by the principal component axes by computing the component scores (coordinates in PC space) for each sample. The component score matrix S was given by:

$$S = XB \tag{2.1}$$

where X is the data matrix, standardized with mean 0 and standard deviation 1, and B is a score coefficient matrix such that:

$$B = AC - 1 \tag{2.2}$$

A is the component loading matrix, and C represents the variance-covariance derived from the retained components, given by:

$$C = A^T A \tag{2.3}$$

The component scores could be calculated with Equations (2.1), (2.2) and (2.3). Every sample was projected into the new four-dimensional Kansei space in this way.

2.4 Artificial Neural Network (ANN)

The most important component of this system is the neural network. It is responsible for accurately mapping the Kansei-space to the shape-space spanned by the shape parameters. A properly set up and trained neural network has the ability to generalize, that is, produce accurate outputs for inputs not encountered during training. However, it is difficult to achieve good generalization, and in order to make a reliable system it is essential to make sure that the network solution is accurate by validating it.

Generalization is influenced by the size and quality of the training set, the architecture of the neural network, and the physical complexity of the problem. There is a trade-off to be considered regarding the training set - Design of Experiments (DOE) [42] will minimize the samples and make the survey easier to perform, but at the same time the neural network training will benefit from a larger dataset. With a fixed, small training set and no way to control the complexity of the problem, the network architecture was carefully chosen to able to represent the underlying problem and achieve a good generalization.

A feed-forward, back-propagating neural network [20] was constructed with four nodes in the input layer (image-space), and 12 nodes in the output layer (shape-space),corresponding to the shape parameters for the side profile shape.

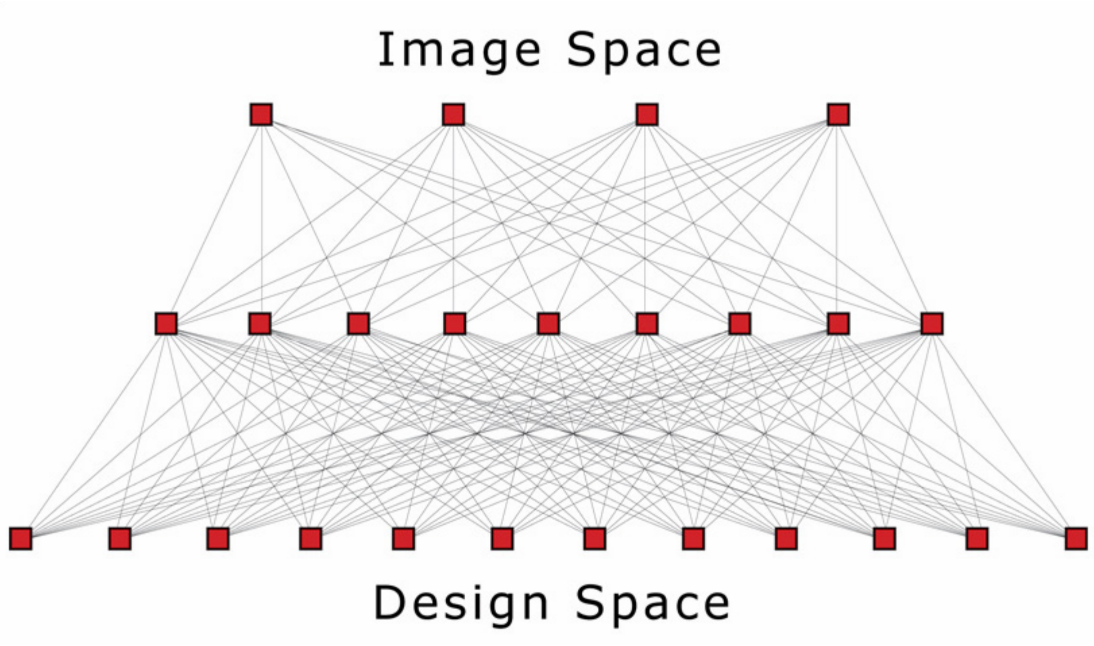

Figure 2.6: Neural network model

This network uses gradient descent on the errors to train the weights, with a differentiable activation function of the weighted sum of inputs v, defined by:

$$\varphi(v) = \frac{1}{1+e^{-v}} \tag{2.4}$$

As this sigmoid binds the output in the interval 0 to 1, it was necessary to normalize the design parameters for the samples in order to use them as the desired output, or target values, for the training.

With a hidden layer of nine nodes, a 4-9-12 structure of the neural network was used for the training of the side profile shapes. To avoid over-training and losing the ability to generalize, error decay was implemented into the network, while a small learning rate and momentum provided stable and efficient learning. The number of hidden units was a critical parameter of the network – too few units in the hidden layer will not give the network enough flexibility to properly represent the unknown underlying function, whereas too many units may lead to a network that also fits the noise, not just the signal, leading to over-fitting. Neural networks trained with a scarce amount of cases are prone to over-fitting, and validation must be performed to test the performance of the network. The cross-section profile shapes were separately trained on a different network architecture, with a dataset based on only four samples. Due to the limited data available, the use of a test set would waste a lot of data for the training and therefore multi-fold cross-validation methods seemed appropriate for this system, as all the cases can be used in the training. Leave-one-out cross-validation was performed to estimate the performance of a number of network models, and aid in the selection of the best model. This method was also implemented to incorporate early stopping in the training. By taking these measures, it was possibly to achieve a neural network with good generalization, despite the small training set.

2.5 Shape generation

The design parameters were given by the neural network solution, and therefore a 3D-model of the desired car could be constructed. The aim of this system, programmed in OpenGL^{TM}, was to generate a simple model and display the proportions suggested by the neural network solution. In order to build a curve for the side profile shape, a series of seven Bézier curves [12] of order four were joined together to form the mid-line, side profile curve in Figure 2.7.
This curve was duplicated, transposed along the z-axis and had some control-points altered by the cross-section profile parameters to form the shoulder-line curve. After these two curves were laid out, Bézier surfaces were used to connect the curves and create the surfaces for the model.

Bézier curves pass through their endpoints, which allowed the endpoints for the side profile curve to be placed according to the solution presented by the neural network, and thereby yield the desired proportions. Furthermore, one desirable feature of this type of curve is that it allows local changes of control-points without affecting the shape of the whole curve. Therefore the joints were restricted to tangent continuity (C1), as displayed in Figure 2.8. The interior control-points also influence the shape of the curve, but as this work focused on the basic proportions, these points were only used to ensure C1-continuity and to give the curve some smoothness. Thus the variables dx and dy were kept at a small constant and considered to have no effect on the form impression.

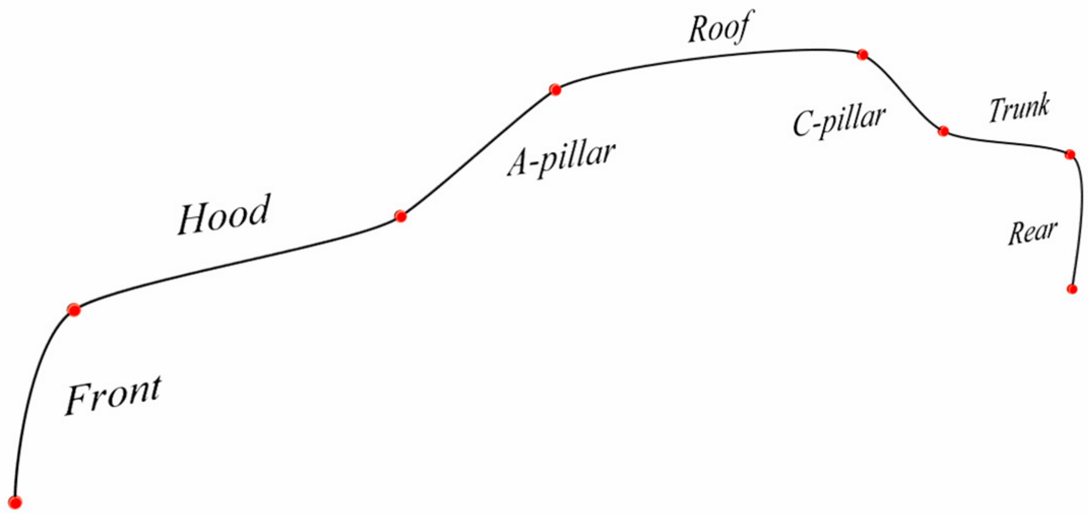

Figure 2.7: Side profile curve

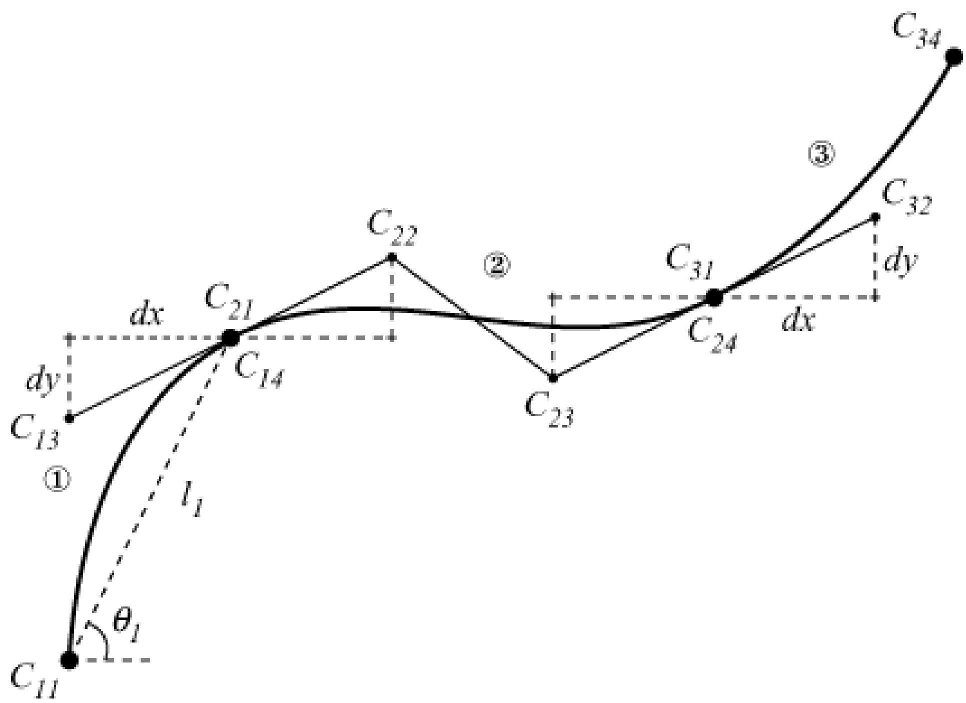

Figure 2.8: Bézier curve

The results obtained from this method depend on quality data from the survey and good performance of the neural network. If either one of these requirements fail, the neural network will provide a solution that is not correct – it is therefore essential to validate the correctness of the data and the neural network solution by the methods mentioned in §2.4.

A visual inspection of the suggested model can give some confidence to whether the solution is correct or not, but it should not be trusted without prior validation of the network. The results of entering the component scores of four different sets of form impression into the neural network are presented in the figures below.

The "average" car has a form impression vector where the weighted attributes are all set at 3, like:

$$\mathbf{F} = \begin{pmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{pmatrix}.$$

By feeding this form impression to the system, the model in Figure 2.9 was generated.

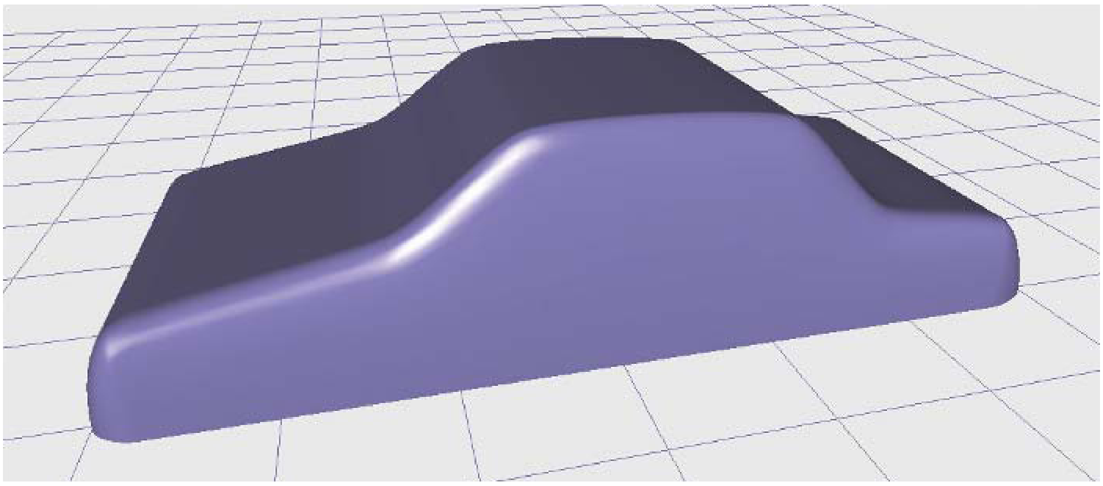

Figure 2.9: Average side profile

Increasing the Cute parameter to the maximum value 5 and keeping the rest at 3, like:

$$\mathbf{F} = \begin{pmatrix} 5 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{pmatrix},$$

yield a model with more playful and cute features.

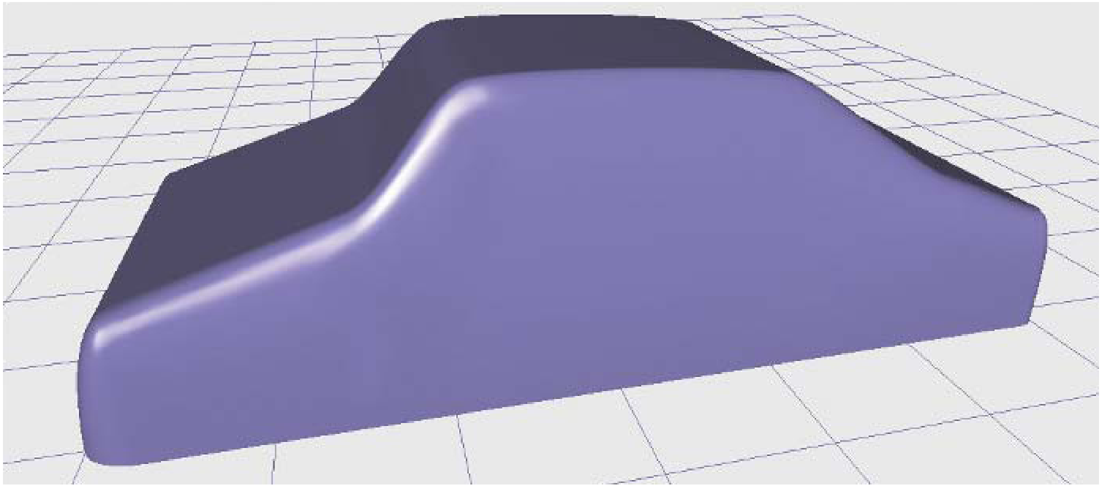

Figure 2.10: Cute side profile

When parameters for Robust and Spacious are increased to 5:

$$\mathbf{F} = \begin{pmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 5 & 5 & 3 & 3 \end{pmatrix},$$

a noticeable change in the angles of the A- and C-pillars, which allows for maximum space in the glasshouse, can be seen. The rear of the car has also undergone a few changes to give a more robust and spacious impression.

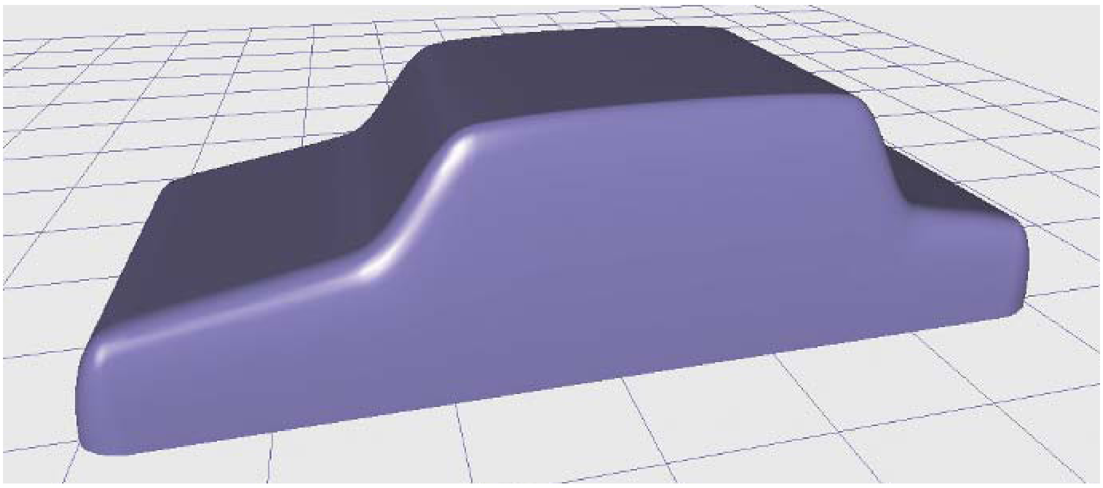

Figure 2.11: Robust and Spacious side profile

The result of maximizing the parameters for Sporty, Modern and Sleek, like:

$$\mathbf{F} = \begin{pmatrix} 3 & 5 & 3 & 3 & 3 & 5 & 3 & 3 & 5 & 3 \end{pmatrix},$$

yield a model which seems to carry those features. The A-pillar angle is very small, and the glasshouse is shifted to the rear, with a sleek front.

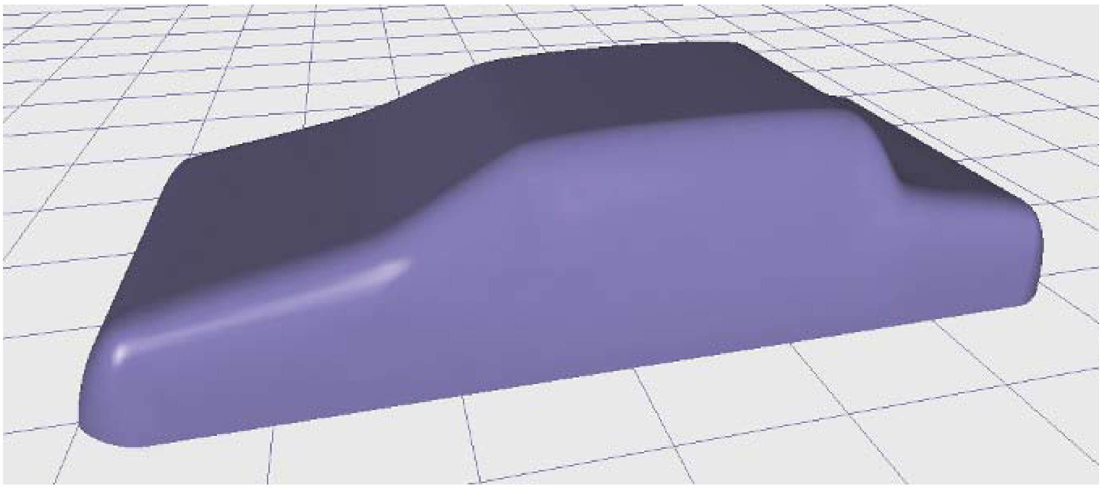

Figure 2.12: Sporty, Modern and Sleek side profile

There were only two design parameters established to describe the cross-section profile shape (see Figure 2.3). Variations in tumblehome and shoulder-width evoke different form impressions, as evident by the figures below. The side surfaces have been colored grey to clearly distinguish how these two parameters change the model. The result for the “average” case, that is:

$$\mathbf{F} = \begin{pmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{pmatrix},$$

is displayed in Figure 2.13.

Considering the same case as in Figure 2.12, where the Sporty, Sleek and Modern parameters had been maximized, the cross-section profile shape is now altered with a wider shoulder-width and a larger tumblehome, which seem to increase those features. For the Robust and Spacious case, it can be seen that the increase in these parameters has lead to a tumblehome that is very small, in order to provide maximum space inside the car.

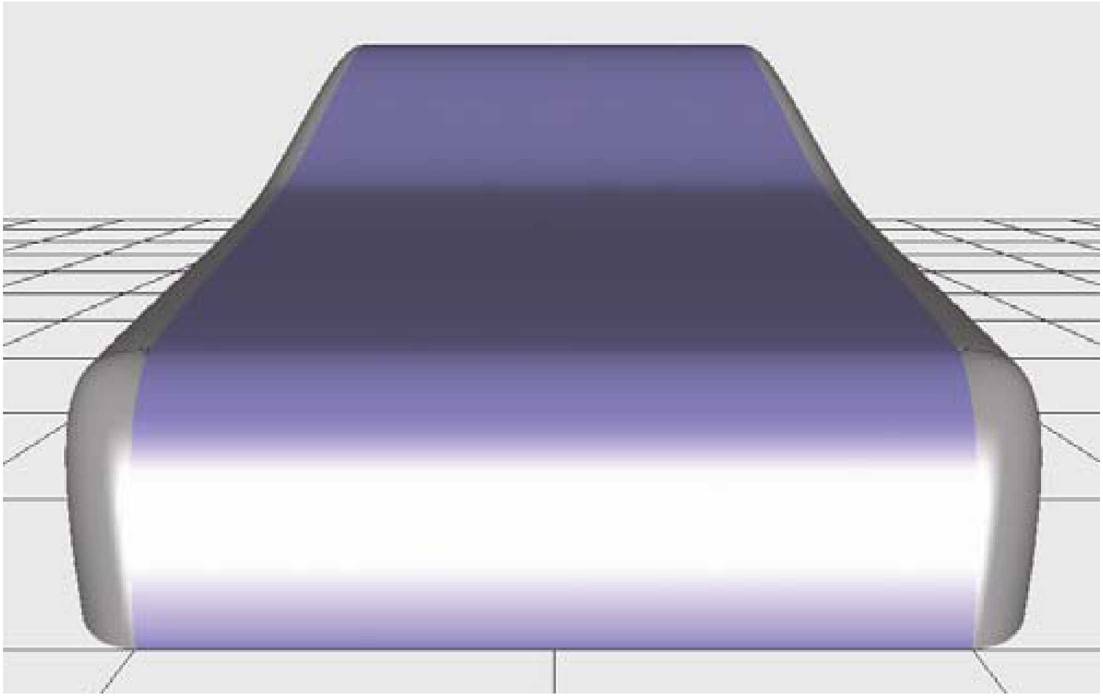

Figure 2.13: Average cross-section profile

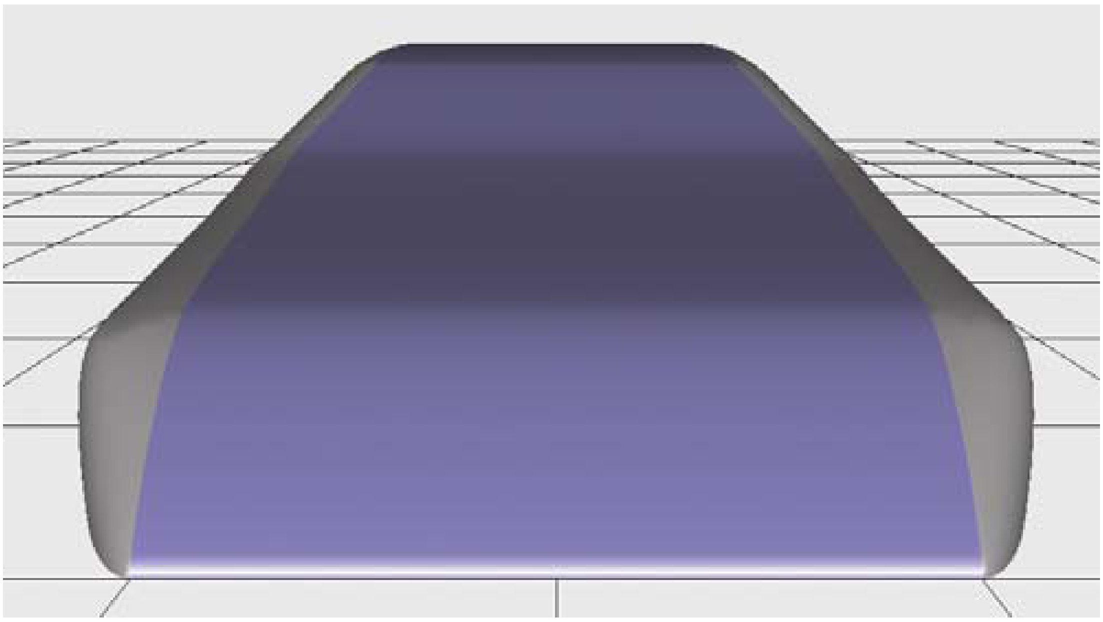

Figure 2.14: Sporty, Modern and Sleek cross-section profile

A designer can explore shapes in this manner; by moving along different axes in Kansei space, he or she can get an intuitive feel for which design features evoke a specific emotion. Although these 3D-models are simple, they can still be used as a starting point or foundation for different design concepts modeled to a higher specification.

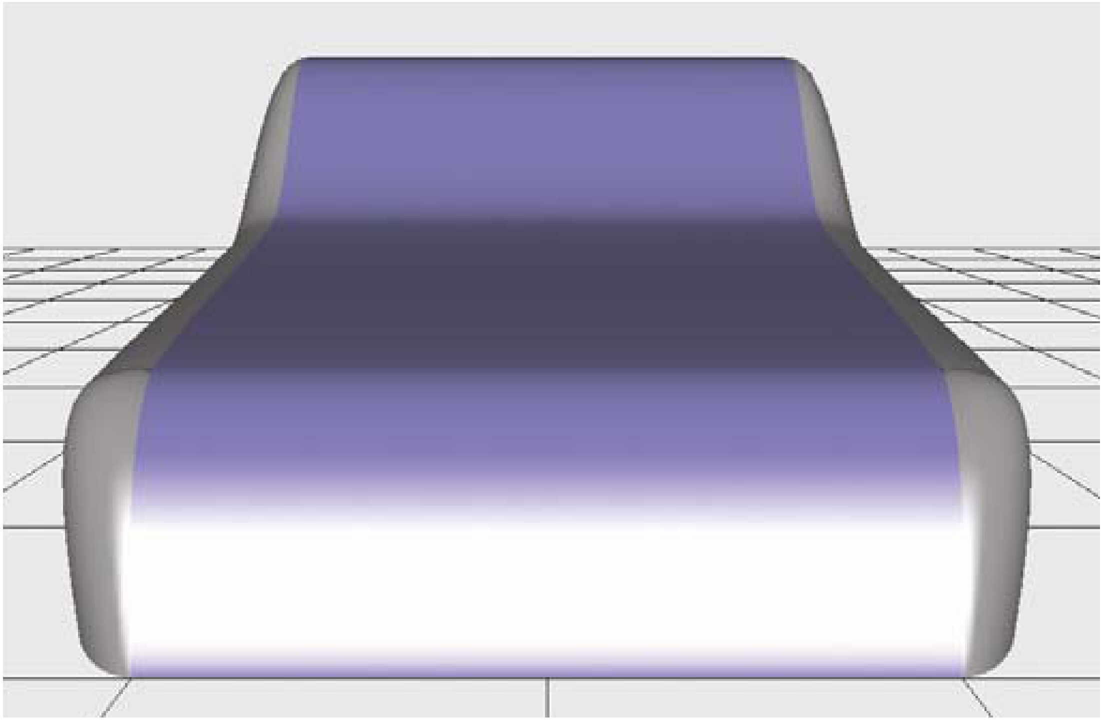

Figure 2.15: Robust and Spacious cross-section profile

2.6 Chapter summary

To conclude the methodology presented here, a 3D-model of a car was created from shape parameters, which were calculated by a neural network trained to relate subtle form impressions to basic proportions of a silhouette line. Validation of the neural network performance and a visual inspection of the models created by the system indicated that the system could produce accurate results. This shows that it is possible to create a design support system that can aid the designer in the creative process by suggesting shape parameters relating to a specific form impression.

It is also possible to teach the neural network the reverse mapping, that is, to have design parameters in the input layer and form impressions as an output. In this case it would be possible to verify the form impression based on a specific shape. This procedure could be very useful in an iterative design process with frequent re-designs, where it is important to make sure that the shape modifications are moving towards the desired Kansei.

These results are promising for future research in this methodology. Applications are not restricted to only proposing basic proportions of a car, like in the study presented in this chapter – the methodology can be modified for use in most studies where the mapping between Kansei and a parameterized artifact is sought. However, one caveat of this system is the fact that form impressions are very complex, and not only associated with a few elements of a product. For a car for instance, the color, material, sound, details in the design and even the marketing to promote a specific image may feed into the perceived Kansei. Thus the extraction a few parameters from such a complex process is a simplification of this problem. This is a necessary simplification, as a large set of parameters would lead to a survey with so many samples that it would be virtually impossible to carry out in an efficient manner. The problems of this type of simplification, which include the representation of real artifacts and reduced affective bandwidth, are the topics of the next chapter – Kansei Fidelity.

Chapter 3

Kansei Fidelity

Kansei Engineering (KE) methods need accurate data in order to build models to explore the intricate relationships between an artifact and its associated Kansei. However, many domains will not, for practical and economical reasons, allow the use of real artifacts to measure Kansei. Instead, representations of the real artifacts, such as sketches, images, Virtual Reality (VR) and scale models are used in surveys to collect data. The problem with this approach is that Kansei measured with representations might not necessarily be the same as the true Kansei evoked by the real artifact. A study by Ferwerda et al. [13] showed that shape discrimination was affected by viewpoints and rendering-techniques used to represent a design concept in 3D. Likewise, the aim for the study presented in this chapter was to investigate if measured Kansei had these types of discrepancies between representations of an image and a physical model. That is, what role will a restriction in sensory inputs play during an exploration of a design concept?

This chapter outlines the problems with representations, and how poor presentations of an artifact can bring havoc to the results collected in a survey. A pilot study is presented, where two different representations, consisting of images and 1:18 die-cast scale models, of two different cars (BMW M6 and Mercedes CLS 500) are used. These representations of cars were evaluated on six Kansei-words by a group of subjects. The results show significant discrepancies between the representations for the Mercedes, but none for the BMW. This indicates a difficult challenge for accurate models of complex artifacts in KE - if there are patterns in these discrepancies that are directly associated with the limitations of affective bandwidth of the representation, a set of weights to correct measured Kansei can be calculated easily. This study proved the existence of discrepancies between an image- and scale-model-representations, but did not discover any patterns that could be used to correct the inaccuracies of these representations.

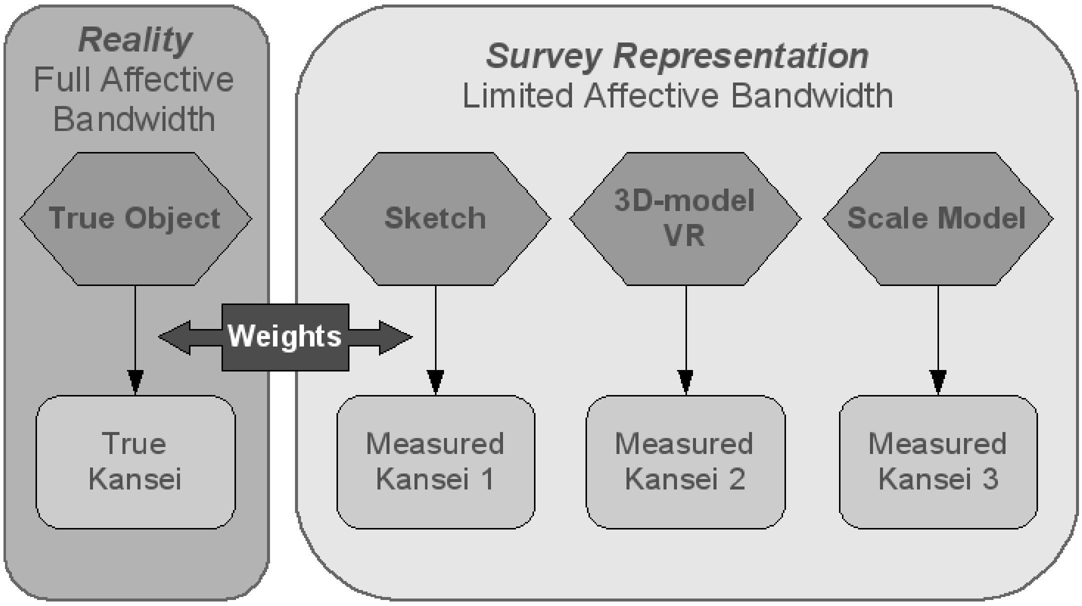

Figure 3.1: Survey representation of real artifact

A "true" object is a real artifact, with no simplifications or limitations in features. We can employ all our senses to explore such an object; there are no limitations imposed unless they occur naturally in the domain of the object. That means that all features and properties of the object can communicate with a person via a full affective bandwidth:

> *A scent of leather greets me as I ease my way into the driver's seat. The closing door cuts the noise of traffic outside with a solid thump. I take a moment to adjust the seat and reach out to grip the steering-wheel. Perfect. Mirrors OK. My hands treble ever so slightly as I turn the ignition-key and feel the engine come to life. No drama yet, only smooth power as I navigate through the streets of Funabashi and paddle-shift through the gears. Then suddenly, an open stretch*

of road emerges. I stop and push the M-button, which remaps the settings from timid to brutal. A rev-counter pops up on the windscreen. The deep rumble from the exhaust sends shivers through my spine, and I am transformed into an F1-driver, blipping the throttle just to get a preview of what is to come. I take a deep breath, push the gear-lever into launch control, and floor the accelerator...

Kansei is an emotional response triggered by an external event or artifact; it is therefore inherently difficult to externalize and measure it with any degree of confidence. We try, through careful definition of spaces of both physical properties and semantics, to bring structure and create models that will bring clarity to these intricate relationships. These models will, however, not yield accurate results unless they have accurate data to start with. Data is usually gathered in the early stages of a design process, when only rough representations of concepts are available.

Many domains, such as architecture and automotive, must depend on sketches and VR throughout most of the design process, and certainly for surveys, due to the very nature and scale of the domain. It is just not economically feasible to build fully functional prototypes for Kansei measurements in a survey. Therefore, we resort to simpler alternatives, with limitations in affective bandwidth, and present a design concept through sketches, images or 3D-renderings.

But, are these representations directly applicable to the true object? Is it plausible to collect data with an image and expect it to hold true for an object we can touch, smell, hear and interact with freely? These questions urged me to undertake the study presented in this chapter.

The aim of this study was to investigate if there are significant differences in measured Kansei of an artifact, that can be directly attributed to the representation used in a survey to collect data. If there are any patterns in these differences, it would be possible to create a set of weights to correct the discrepancies between reality and survey, and thereby present a more accurate foundation, from which models can be built upon.

3.1 Survey

A pilot study was conducted in order to investigate if and how two different representations of real artifacts affect the perceived Kansei of a group of subjects. My research is related to the automotive field, so I chose to use two premium cars for this survey, as described in §3.1.2.

3.1.1 Subjects

21 subjects took part in this study. They were predominantly Japanese male in their early 20's, none with a particularly strong interest in cars. Everybody was familiar with the general brand-image of the chosen vehicles and recognized them as premium, but none knew the particulars or detailed specifications, as I found out through interviews before and after the survey assessment, which was conducted unsupervised.

3.1.2 Artifacts

For this survey the Kansei of a BMW M6 (E63) and a Mercedes CLS 500 was measured. These cars belong to the very exclusive end of the premium segment, and offer both luxury and performance in the same package. However, to people without a strong interest in cars, they have relatively few visual features that will give an indication of their sports-car abilities and massive engines.

Both BMW and Mercedes have strong market-shares in Japan, and their respective brand-images are well-known. In order to avoid any bias due to color, the cars were chosen with similar colors. These two cars were presented in the survey via two different representations, as described in the following sections.

3.1.3 Image representation

The image representation shows the car from front and rear-view, as depicted in Figure 3.2 and 3.3 below. The size, resolution and background of the images were kept constant for both cars. The image did, naturally, only allow subjects to explore the artifacts by vision on a computer-screen. This is a natural interface for this group of subjects, but nonetheless severely limits the affective bandwidth.

Figure 3.2: Mercedes CLS500, front view (photo courtesy of Mercedes Benz USA)

Figure 3.3: Mercedes CLS500, rear view (photo courtesy of Mercedes Benz USA)

3.1.4 Die-cast scale model representation

As a second representation, I used die-cast models in scale 1:18, purchased from the Kyosho company [26], which specializes in very detailed and accurate models for collectors. These models are not toys; they represent the real cars very accurately, thus allowing subjects to explore them with both eyes and hands. The doors, bonnet and trunk can be opened and give an idea of the interior of the car. Compared to images, it is much easier to understand the exterior shape, details and proportions by using these models. The BMW (shown in Figure 3.4 below) has product-number KO8703BZ and the Mercedes is identified by product-number KO08401DGY.

Figure 3.4: BMW M6 die-cast 1:18 scale-model, next to pen

3.2 Kansei words

The Kansei language used in this survey was intended to describe cars in the premium segment, and therefore only six words (see Table 3.2) were selected by browsing company websites and magazines portraying exclusive cars. This is of course not enough to properly span the space of premium cars, but the focus for this study was mainly to investigate if and how Kansei change when different representations are used, not to provide an in-depth analysis of the premium car domain. This small set allowed me to interview subjects and collect data quickly.

Table 3.1: Six Kansei words for premium segment

Luxurious	**Classic**	**Modern**	**Spacious**	**Elegant**	**Sporty**

3.3 EvokeDB-system for automated surveys

EvokeDB (Evaluation and Verification of Objects in Kansei Engineering via Database) is a system I developed [35] to automate many of the difficult or tedious tasks needed to create and launch on-line surveys for gathering data. This system is described in detail in Chapter 4 and I will therefore only give a brief overview here.

EvokeDB is a web-application with a database back-end where user queries are handled automatically through an intuitive interface enabled via the Ruby-on-Rails framework. This makes communication, alterations and maintenance of an advanced database structure accessible to users without extensive knowledge in programming. It is possible to define and instantiate own objects such as artifacts, words, realms and senses, and create and launch surveys with a few mouse-clicks.

Furthermore, EvokeDB will organize and store data for any number of different projects with multiple surveys - its structure is very flexible and gives a user control over the data. The most characteristic feature is, however, the ability to create and analyze relationships within the data, thus making EvokeDB an interpreter of Kansei that can be referenced throughout a project.

3.3.1 Survey method and interface

EvokeDB automatically created an on-line survey based on two representations of two cars with six Kansei-words, which gave a total of 24 questions presented to each subject. For each representation of a car, a web-page was constructed. The layout of this page consisted of sections of instruction, presentation, explanation and evaluation. First of all, the instruction clearly stated how to use the interface to evaluate an artifact. This section also explained the range used in the survey, and how to enter an evaluation of each Kansei-word. The presentation-section was dependent on the current representation - for images, each car was portrayed from front, side, and rear by images that could be increased in size so that even small details in the car was visible, whereas in the case of die-cast models, the subjects were given a physical model and encouraged to explore it by opening doors and viewing it from many different angles.

A section with all the Kansei-words followed after this presentation. A slider with a value from zero to ten was attached to each word, and by adjusting it subjects could easily enter their perceived Kansei very precisely. The order of the cars, and the words as well, were randomized to avoid bias. I believe this randomized order also helped combat fatigue in the subjects to some degree. There were only 24 questions but for longer surveys there is definitely a concern that the subjects will start to rate artifacts based on a pattern, unless randomization is implemented. Furthermore, if there was any confusion regarding the meaning of the words, a dictionary entry was made available simply by clicking on the word.

This layout seemed very easy to understand and use for this group of subjects. Before and after the survey there was also a brief discussion with each subject, where the words and survey-interface were explained. I verified that everybody had successfully entered their Kansei for the cars but also noted that some expressed unfamiliarity with the range used. A range from 0 (not at all) to 10 (completely) is used in many different contexts, but several subjects found this rating peculiar and had anticipated the more common 5-point scale with radio-buttons.

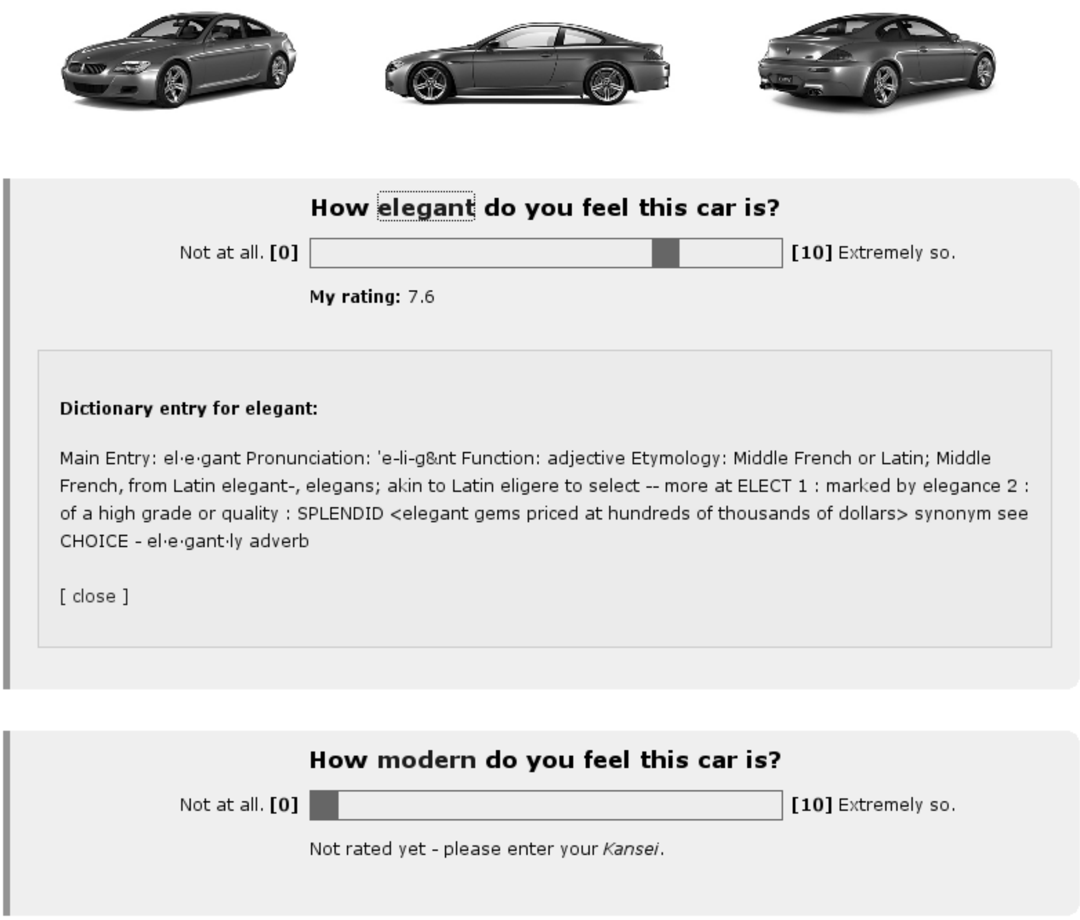

Figure 3.5: EvokeDB graphical user interface (GUI)

3.4 AJAX and on-line surveys

On-line surveys have a number of properties that make them usable for Kansei Engineering studies. First of all, they make large-scale collection of data possible at a low cost. Surveys can be launched on a laptop, on a company Intranet or on the Internet. Furthermore, once properly set up, this type of surveys allow subjects to connect remotely without installing specialized software since most modern computers are already equipped with web-browsing capabilities. Data handling is very efficient if they are connected to a database, as SQL-queries can produce nicely formatted output quickly.

There are, however, some serious drawbacks with web-applications, which up until now have left subjects in on-line, unsupervised surveys less supported in terms of how the system communicates with them. A traditional desktop application is very responsive - if a user clicks a menu or makes an input to the application, the result is displayed at once. On the other hand, the Internet has been, and still is, a far more constrained medium in terms of how we interact with it. Basically, we are presented with content in a hypertext-system which we navigate through by clicking on links that tells a remote server to serve us new information; it is an interaction model of constant call-and-response that keeps users waiting and thus lacks the responsiveness of a desktop application.

AJAX (Asynchronous JavaScript and XML) presents a solution to this problem. AJAX is not a technology in itself, but rather an umbrella term representing a group of technologies and how they are used together. *XMLHttpRequest* is one of the most important components of AJAX as it allows an object to exchange data asynchronously with a web-server through an independent communication channel between a web-applications' client-side and server-side [47]. By implementing AJAX it is possible to create a user-interface that communicates with the user in "real-time", without the constant need to submit data and reload the whole page to get a response back from the server.

For example, a common task in a survey to collect Kansei-data is to present a subject with an artifact, which is then evaluated by the semantic differential method [36] on a set of words (or rather, bipolar pairs of words) representing Kansei. This evaluation is usually performed by making a correlation of how strongly the artifact relates to the word on a range from low to high, either by entering a number, clicking on a radio-button, or adjusting the value on a slider. In a supervised survey a subject can always ask the interviewer about the input method or the meaning of the range, but this is not possible for large-scale, on-line surveys. It is therefore important to give clear instructions and explain the meaning of a range before the survey, as well as provide feedback during it. With AJAX, the system can respond to a users' input directly through a number of predefined events, and thereby provide support and make sure that the meaning of an input is properly understood. A traditional approach on the other hand, is bound by the limitations of the hypertext interaction model and can not cater for this type of feedback during an evaluation.

Two examples of how AJAX was implemented can be seen in Figure 3.5. Initially, the slider is set at zero and there is an accompanying text saying "Not rated yet - please enter your Kansei!". Once the subject interacts with the slider, this text will be changed into "My rating: [slider-value]". This gives a subject direct feedback to the performed action. Secondly, there is the dictionary-function. The dictionary entry is by default not shown as it will clutter up the interface with too much information, but it is still available to the subjects who need it. A simple click on the Kansei-word will present the dictionary-entry through an action that slides out a new section on the page, with no reloading, pop-ups or going back and forth between windows necessary; something that would have been impossible to avoid with a traditional on-line survey without AJAX.

3.5 Results

Once the survey was completed, the data in the EvokeDB-system was exported to other commercial software packages for analysis.

3.5.1 Change in Kansei between representations

Figure 3.6 shows a radar graph of the average value collected for each word in the survey, separated by representations of image and die-cast model, for the BMW M6. Clearly, these two representations evoke almost the exact same response from the group of subjects. Only "classic" shows an increase when the die-cast model was evaluated, but this is a very small increase and the overall pattern would suggest that the two representations can be considered equal.

Figure 3.7 shows the result of the same analysis of the data for the Mercedes. This graph shows significant changes between the representations. For "classic",

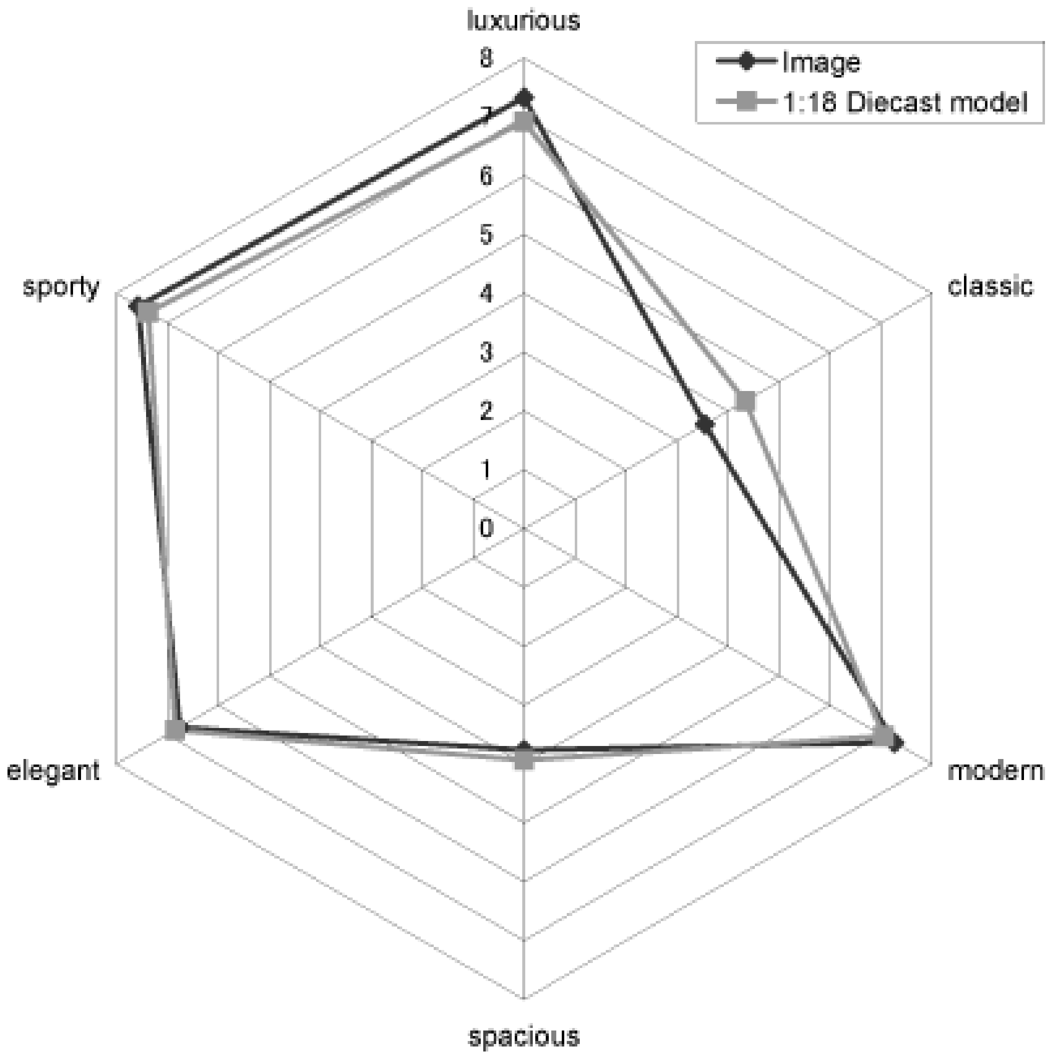

Figure 3.6: Kansei for BMW M6

"sporty" and "modern" there is a very good fit, which follows the results of the BMW-case. However, there is a dramatic increase in "luxurious", "spacious" and "elegant" for the die-cast model representation, with close to 20% higher average values entered for these words. This indicates that the two representations are not equal. Furthermore, comparison of Figures 3.6 and 3.7 reveals that there is no consistent pattern between these discrepancies, which one could contribute directly to the limitation in affective bandwidth between representations.

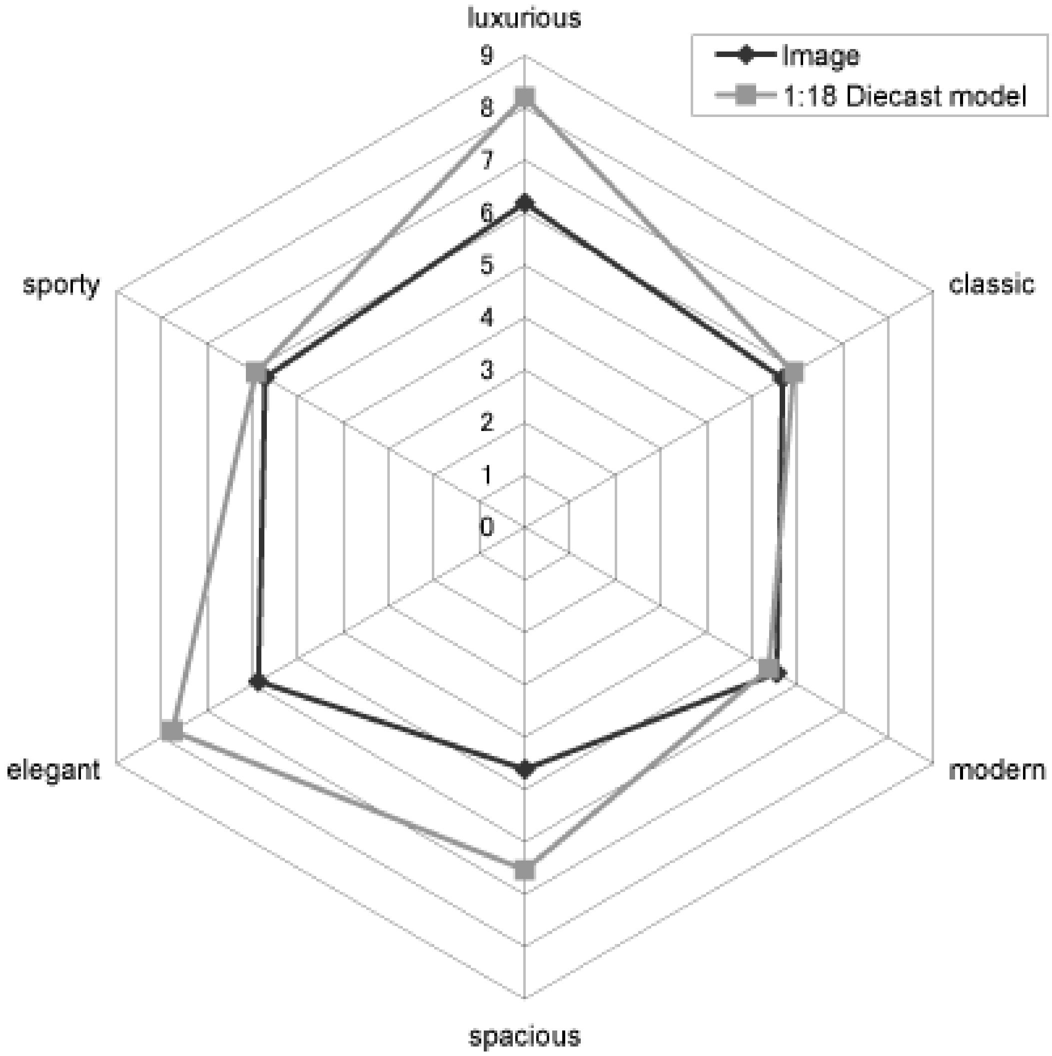

Figure 3.7: Kansei for Mercedes CLS 500

This is really the worst case scenario - I have showed that there are significant changes in measured Kansei due to limitations in affective bandwidth, but if no pattern exists, it will not be possible to create weights to counterbalance these differences. It seems some features of the cars are more strongly affected by inefficiencies in the representations than others. For example, the Mercedes has four doors, and it seems plausible that this particular feature was not properly presented by the images. Only the die-cast model, which allowed subjects to open doors and peak inside the car, could present this feature and evoke a stronger Kansei. Likewise, the image-representation failed to bring out the Mercedes' features of luxury and elegance. At this point I don't have data to correlate these representations to the real cars, although this would, of course, be the most interesting comparison to make.

3.5.2 Distribution of changes

The next step of the analysis was to see if the results in the previous section was due to a few outliers, caused by subjects who completely changed their opinions between representations. Figure 3.8 shows the distribution of relative change in Kansei, for 126 entries (all words) for the BMW M6. This graph shows that most subjects made a rather modest change in Kansei between the representations, and also that the valence of changes was balanced, that is, there was a close to even number of increased and decreased changes in Kansei performed. A majority were within a 20%-change, and only a few subjects changed their Kansei drastically. Therefore these results are not influenced by a few outliers.

For the Mercedes-case (displayed in Figure 3.9), the graph shows that the changes in Kansei were not balanced, with many subjects increasing the values for Kansei-words when the die-cast model was used. This graph is also not smooth, there are several maxima, not only at 5%, but also at around 25% and 55%, which indicates that a significant number of subjects changed their Kansei rather drastically.

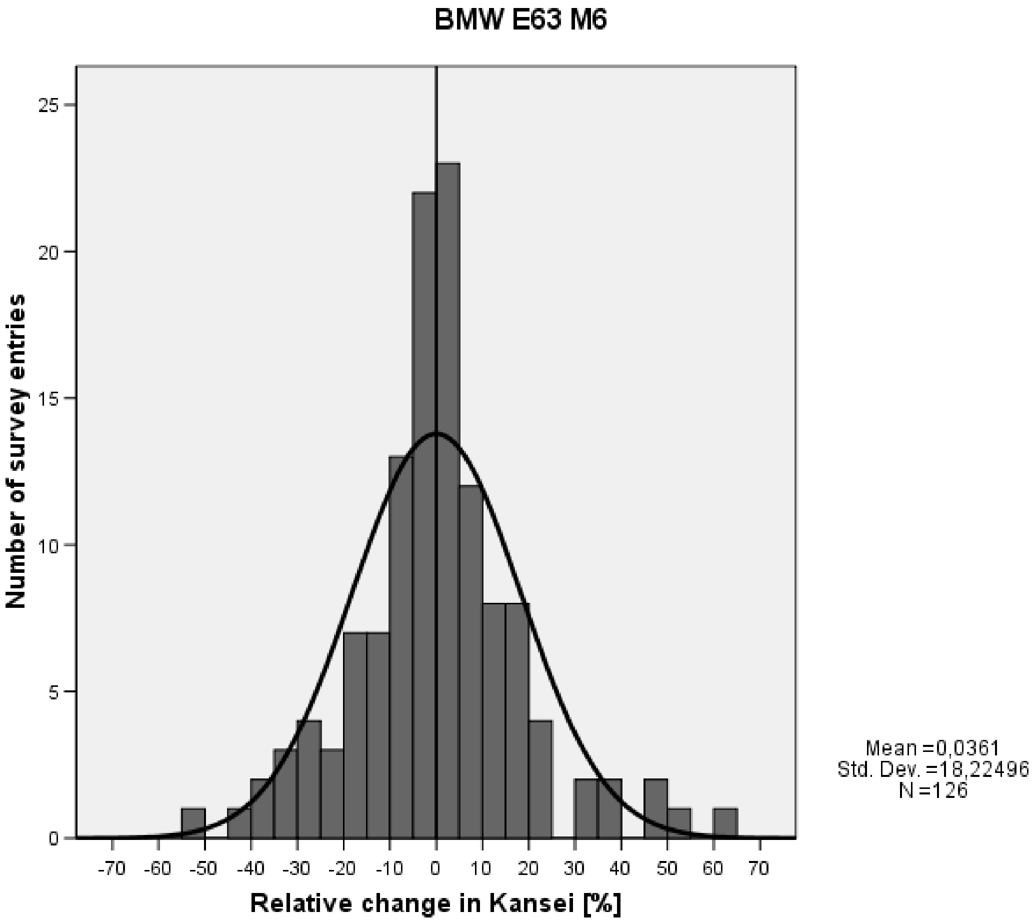

Figure 3.8: Distribution of relative change in Kansei for BMW M6

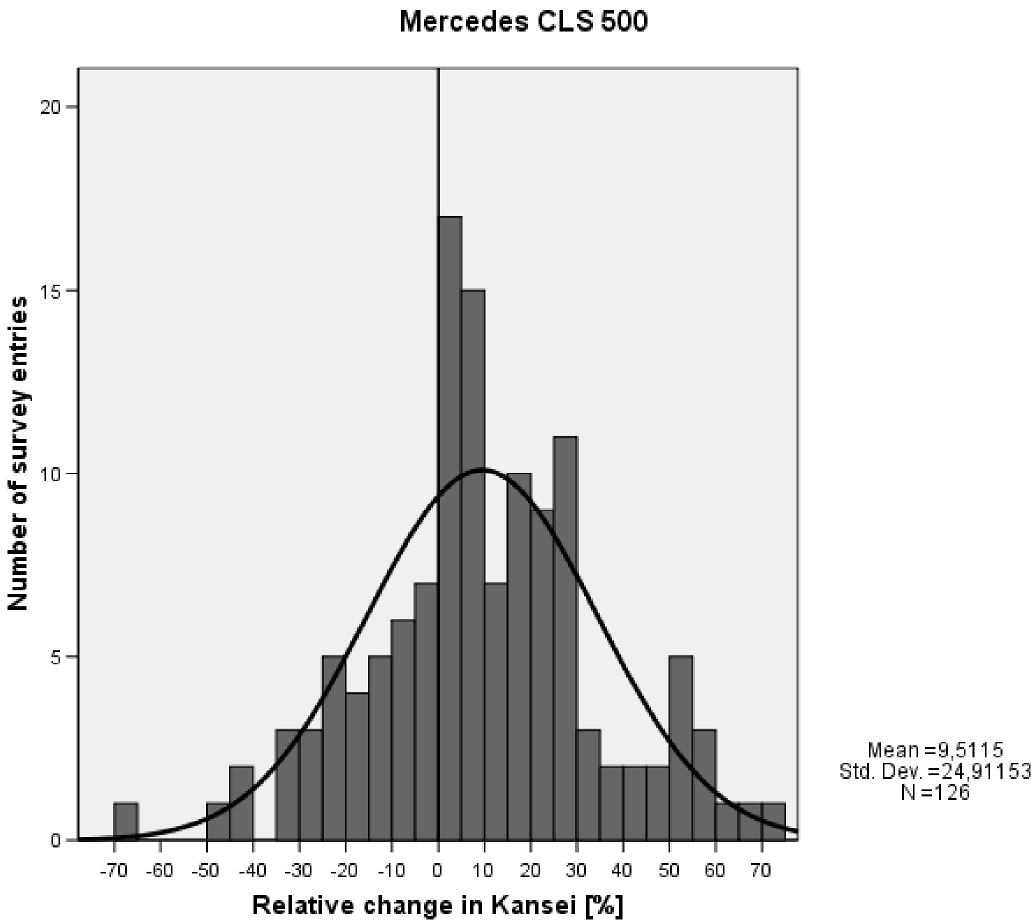

Figure 3.9: Distribution of relative change in Kansei for Mercedes CLS 500

3.5.3 Discussion

So, what can we learn from these results? First of all, if we for a moment consider the die-cast model as the more accurate of these two representations, the distribution of survey entries should have less standard variation compared to the data collected with images. That is, if a representation can accurately present the features of an artifact in terms of Kansei-words, the group of subjects should produce a Gaussian distribution with a characteristic bell-shape, with small standard deviation. However, this was not the case for the data collected in this survey. In some cases the die-cast model would produce less variation in the data, but this was not a consistent pattern by any means, the image-representation also had many cases with less variation (more consensus) in comparison to the die-cast model representation.

Another concern is that the data collected for this study did in many cases show great variations in distribution, and spikes instead of a pretty bell-shape. It is essential to make sure that there is some consensus within a group of subjects, that they all have the evaluated the artifacts based on Kansei words that carry the same meaning to them, otherwise the results will be biased by the duality of meaning. Multiple maxima in the data usually means that there are several meanings of a word within the group, or that the word was not understood. This is not uncommon for groups that are not completely homogeneous, and in this study there were differences in cultural background, as some subjects were recruited among international exchange students.

The most significant factor for this poor level of consensus is probably one related to language. None of the subjects spoke English as a first language, and although everybody seemed to understand the meaning of the words, they did not use these words in their everyday, active vocabulary, neither were they accustomed to using them specifically to discuss features belonging to the domain of cars, and I believe this introduced noise to the collected data. However, the relative change in Kansei between representations should not be strongly affected by this, and therefore I believe that the discrepancies between representations reported in this chapter hold true.

The study presented here should be seen as a pilot study, meant as a first step in recognizing that a problem exists. Studies of larger scale must be conducted, where statistically significant results can prove if there are any patterns in the discrepancies of measured Kansei and "real" Kansei, which can be directly attributed to limitations in affective bandwidth. Such studies should preferably include the real artifact, not just investigate relationships between representations. I am confident that the discrepancies I have presented exists, but I am not comfortable to develop any methodologies to rectify them yet, based on the data from this study.

However, these results still suggests that a pilot study should always be performed when using representations. This way, it is possible to get an idea of the shortcomings of a certain representation, and the results can also identify Kansei words that may yield noisy data – in this case the pilot study could be used to match Kansei words to a specific representation.

3.6 Chapter summary

Many domains will not, for practical and economical reasons, allow Kansei data to be surveyed by the real artifacts. Instead, representations such as sketches, images, 3D-models, VR and scale models are used. However, if these representations cannot properly present the artifact and its design features, it will not be possible to create accurate models based on this data.

The study presented in this chapter shows that there exists discrepancies between data collected with images, and data collected with die-cast scale models. Due to limited data collected, this study could not reveal any clear patterns in these discrepancies that could be directly associated with a particular representation, but I could still identify a problem that is often overlooked, and propose a pilot-study as a necessary tool to identify Kansei words that are not accurately representable by a 2D-image, or a physical model.

Chapter 4

EvokeDB – A survey management based on Ruby-on-Rails

The introduction of this thesis described how Kansei Engineering (KE) is gaining momentum as a tool in product development, and how innovation and knowledge about the customers' desires have always been a key to success in creating products and services that stand out in the marketplace [43]. However, in many cases nowadays, it is no longer enough to be the best performer – customers are often quantitatively satisfied and are not making purchasing decision based purely on logic, but rather on how a product feels; it must appeal to them on an emotional level. Advances in technology, manufacturing and marketing are still essential, of course, but companies must also understand how to use the channel of Kansei-communication in order to survive and excel in a competitive environment. Careful collection of Kansei data from different market segments is therefore crucial; most successful design concepts are born from understanding and meeting not only the basic needs, but also the desires of the targeted customer.

Kansei data is extracted from an internal process that is inherently difficult to come to terms with and analyze with any degree of confidence. Add to this the fact that the presented frameworks (e.g. [10, 17, 31]) are rather general, and that most methods used today stem from generic algorithms in statistical analysis, which require extensive training and system development skills that most designers and product developers lack, or are unwilling to learn. I have covered some of these methods in Chapter 2, but omitted the details of data collection until now. This part is often seen as trivial, because after all, most scientific efforts are based on data collection and such methods are well-documented and taught from primary school. However, as we will see, a system for Kansei surveys can be very helpful and lower the learning-curve for non-experts.

Firstly, it is possible to use a very simple technique in data collection – interview subjects and take careful notes. However the subsequent data entry into a database or spreadsheet is tedious and error-prone, and this approach is therefore unsuitable for quantitative studies that require large amounts of data. This approach will also not be very useful to manage a large number of different surveys for different projects; it will quickly grow too complex, and there is no way to automate parts of the data analysis. In this chapter I will try to present better guidelines on how to describe artifacts and implement methods, so that users without specialized training can carry out successful Kansei-studies.

First, consider complex artifacts, such as most gadgets that cater for our new digital lifestyle - they have a number of stimuli in different realms that may all lead to associations which feed into our Kansei. For example, a portable media-player has stimuli such as shape, material, color, features and services attached to it, and yet there seems to be little consideration taken to how these stimuli are related, and by which senses they are explored, in many KE-studies. I argue that there lies a great danger in overly simplified models for complex artifacts - even a properly performed analysis of a simplified model may yield misleading results, or simply fail to recognize the most important stimulus for a given Kansei.

Thus, I think we can all agree that KE is powerful in skilled hands, but it does presents the novice with a rather difficult set of methodologies that must be mastered in order to implement it successfully. Even a trivial task, such as data collection, may challenge many potential users with a hurdle they cannot easily overcome. Data is a cornerstone of every Kansei-study, and surveys to collect it are essential, but often tedious to create and perform. Consequently, there is definitely a need for tools to automated surveys and guide novices through a study.

The second point I want to make in this chapter introduction is the problem of the reliability of the data itself; is it still valid, can we trust our results? Currently, it takes resources of time and expertise to collect Kansei data, which is often regarded as a beautiful box of sushi (very tasty, but easily spoiled unless you consume it at once). We must find a more robust way to gather, analyze, and store Kansei data to make it more resilient to changes in trends. A red dress may not be "fashionable" every year, but it still a red dress, that is explored by sight or touch, and its stimuli may lead to the same associations in a subject no matter what the current trends dictate. I believe associations are very important to Kansei, as we understand and feel through connections to past experiences. In fact, our whole being depends on the rich set of links that our nervous system is made up by; neurons fire and we *relate*. There is meaning in an object, a color, or a scent, and how they are combined and fused together by our sensory organs to be processed internally. To find this meaning is the holy grail of KE, but the path to Kansei enlightenment has many pitfalls and we need guidelines and systems to support our endeavor. But what type of system could offer this type of support, in terms of accessibility, clarity and guidance? A system for Kansei-explorations must, at some level, possess the ability to relate like us, in order to shed some light on the meaning within the data.

Therefore, I created a database-system that maps the relationships between Kansei-words, stimuli, senses, realms, artifacts and subjects. I believe this can present users with a more detailed and clarified understanding of Kansei, where costly data is never wasted, and new data is easily collected. The system is named EvokeDB (Evaluation and Verification of Objects in Kansei Engineering via Database), and is offered as a tool to make gathering, analysis and storage of Kansei data accessible to non-experts in any field, via automated surveys. My contribution is twofold:

1. Firstly, a new model of Kansei for complex artifacts is suggested, where realms and senses are considered important entities for exploring Kansei.

2. Secondly, I explain how to implement and use that model in a system of open-source components, with functions to create and launch on-line surveys. Specifically, I discuss the merits of the Ruby on Rails framework with

AJAX for this type of application, and how they can be used to create a system that is responsive, flexible, easy to implement, and usable throughout a project as a source of Kansei interpretation.

4.1 System Aim

The introduction pointed out a few areas of KE where I wish to make a contribution. The primary aim of this research project was to create a tool that would be able to support non-experts in any field with means to gather, evaluate, and store Kansei data for artifacts of any complexity. This aim presented me with a number of requirements.

First of all, "non-experts in any field" is a very broad audience with various requirements. Therefore the system needed to be flexible, so that users can make alterations to meet the rquirements of a specific project, whether it is about a new car, a theme-park ride, or cup-noodles. This flexibility should, however, not come at the expense of increased complexity and less usability, as this would discourage users without extensive knowledge of programming computers. This lead me to consider a web-application, which could be developed with open-source components, distributed freely and deployed on different platforms. A web-application would also, once properly set up, allow remote users to interact with the system without installing specialized software, since most modern computers are already equipped with web-browsing capabilities. This would, for instance, enable teams of designers to collaborate, share data and use the system even though they are geographically separated. Surveys could also be launched online, and this feature makes it easier to approach a large number of subjects for a low cost.

However, even though web-content can be created quite easily by many applications nowadays, it does in general take skills and time to create a working survey backed by a database. This system should therefore have functions to automate many of the tedious parts of creating and conducting surveys, thereby making them more accessible as a tool in KE.

Furthermore, I wanted to implement a robust structure where data from many different projects could be stored and shared. This structure should be based on a model of Kansei that considers the complexity of products and services, how relationships between fundamental characteristics affect Kansei, so that designers can interpret data from many different viewpoints throughout the design process and gain a deeper understanding of the meaning of Kansei. It should be easy to run surveys, or consult the system on a word, stimulus or sense, and receive some feedback and guidance based on the data collected.

My focus was therefore initially only on automated surveys, but I also wanted to be able to augment the functionality of the system in the future to include features such as automated generation of Kansei languages and more advanced analytical methods. The model in §4.2.1 was defined with these requirements in mind, although this chapter will mainly describe EvokeDB from the perspective of gathering data.

4.2 EvokeDB foundation

4.2.1 A new model for Kansei

In the previous section I have mentioned a few entities that deserve a more in-depth description. A fundamental problem that I wanted to approach with this system is how to explore the meaning of artifacts, that is, to understand how Kansei is evoked, and to find out what stimuli or events lead to the formation of an emotional response in a person.

What exactly are the main entities we are dealing with? An artifact (product, service or event), is explored by a person through senses (sight, hearing, touch, smell and taste). Theses sensory signals are processed internally to form both a logical understanding and an emotional response, to the artifact. However, the emotional response depends on the environment and context the artifact is presented within, and to a higher or lesser degree, the parts it consists of. Most physical objects have a number of different properties or characteristics, which may lead to associations that feed into a persons' Kansei. The same is probably true, although less obvious, for services as well. I propose an approach where an artifact can be viewed as a whole, but also as an entity made up of parts. To enable this, I introduce the concept of realms.

A realm is a grouping or set of similar characteristics of an artifact. For example, a car could have realms of color, shape, sound, material, or even less obvious realms such as brand-image and showroom, freely defined by the designer. Each realm contains stimuli, such as { red, green, blue, . . . } for color, or { titanium, aluminum, leather, . . . } for material. Inclusion of these entities into the model will allow for a better foundation for the database, as it allows advanced algorithms to investigate relationships between Kansei, senses, realms and the artifact. This is an extended use of Kansei data that will allow for sharing between project in various domains, as the data relates to fundamental characteristics (stimuli) rather than only referring to an artifact in a narrow sense.

Nagamachi [31] explained how the expression "domain" could be used to structure products based on their type, but this should not be confused with realms. Realms are used to structure properties of an artifact, not to describe its type. In my opinion the domain is mainly involved in the logical understanding of artifacts, that is, how they are grouped in a context of other artifacts. That is not to say it has no bearings for Kansei; in fact, it is very important, but it should be supplemented by realms to provide means to explore how fundamental stimuli, like color or sounds, can affect Kansei.

The model in Figure 4.1 depicts entities, in a context provided by a domain, connected with arrows to symbolize relationships. The entities were modeled as objects, along with all their attributes and relationships, to give a strong, yet flexible, foundation to the EvokeDB Kansei database-system. This model allows a user to query the database in a more focused manner, or even to determine how a certain sense contributes to Kansei. In many cases a designer wants to "tweak" the parameters of the artifact to elicit just the right Kansei, but to do so requires a high sensibility to the problem at hand - for instance, changes in one parameter may produce the desired response for a Kansei-word *sporty*, but also bring with it an undesired change in *robust*. The relationships in this new model can guide a designer to the best realm or sense to work within, based on a specific Kansei-word.

Another problem that will also benefit from this model is automated generation of Kansei languages. In most studies it is important to choose a set of words (a vocabulary, or what I will refer to as Kansei language), which can adequately

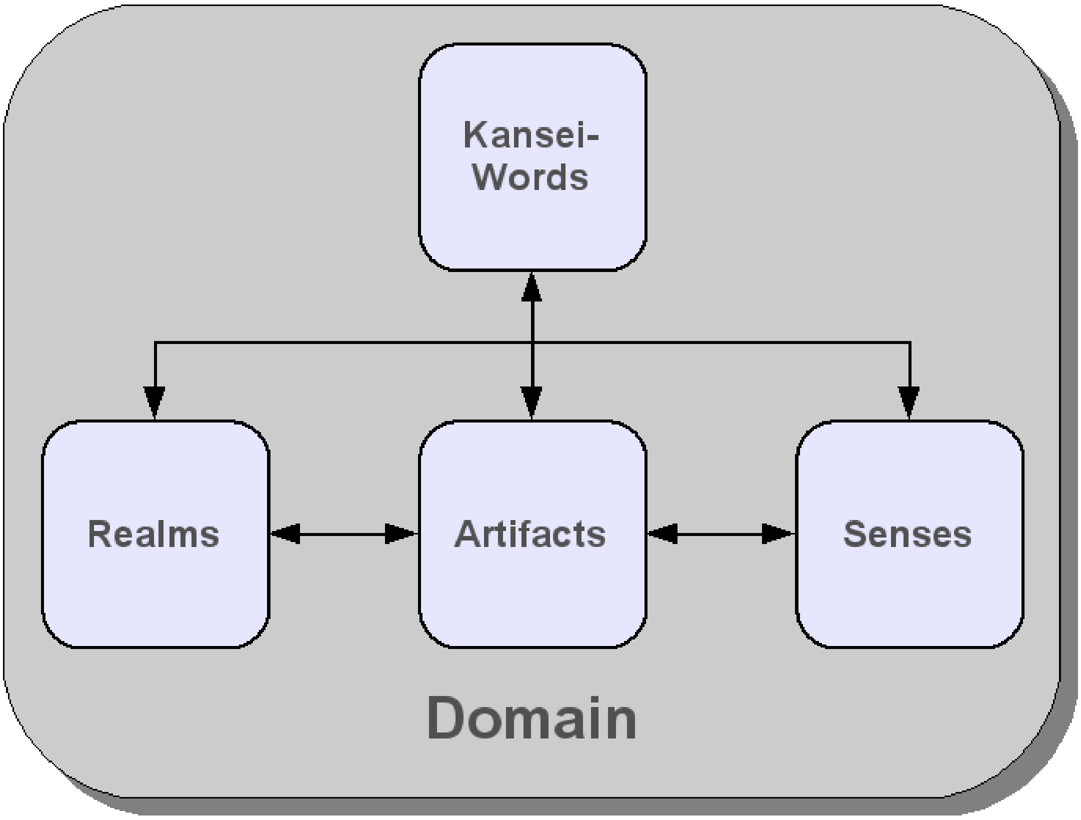

Figure 4.1: Kansei model

describe an artifact and span the space of emotional responses of a group of subjects. There are few guidelines to help a designer select these words – a standard approach is to gather all words that are even remotely related by browsing magazines or manuals, interviewing people, and so on. This leads to a large set of words which can be reduced by removing overlap of meaning by performing a principal component analysis on data from a pilot study (see §2.3). However, this is again very costly and time-consuming, and leaves a lot to be desired in terms of clarity and guidance. Through this new model, it is possible to query the database on which words are most strongly related to an artifact, realm, stimuli, sense, or combination thereof, and propose a Kansei language based on the results. For example, if *red* and *aluminum* will be characteristics of a new product, this knowledge can be used to query the database on data from past surveys to find out which words correlate most strongly to these attributes. As long as a designer can define which realms, stimuli and senses will be used in new project, this feature of EvokeDB will at least give a good starting point, or a base vocabulary, for the designer to consider. Do note that the performance of EvokeDB in this regard will depend on the amount of data available; only a large dataset will provide meaningful Kansei-languages based on past projects. Fortunately, EvokeDB can easily store survey-entries for numerous projects, thereby providing use for old data as well.

4.2.2 Definition of objects

The objects in Figure 4.1 lay the foundation for EvokeDB, but the system also implements a number of additional objects (described below) that were needed for a richer set of functions.

Artifact is a product, service, event or other entity which will undergo a Kansei-evaluation. An artifact can be divided into many realms, like color, shape, sound and material.

Realm is a set of unique characteristics of an artifact. The realms and the stimuli which belong to the realms, will be used to describe an artifact in detail.

Sense describes a human sense associated with a sensory organ.

Subject is a person taking part of a survey. There are many attributes that can be used to describe a subject in detail, such as age, gender, nationality, income level, education, hobbies and so forth.

Survey contains an evaluation on a set of artifacts, on a set of words (language), by a set of subjects. Each survey belongs to a project, and has one language. Different surveys may use different input methods and ranges, so to be able to compare and use data from many surveys, the data must be normalized. The attributes `min_value` and `max_value` is used for normalization.

Language is a set of words used to span a Kansei space. A language can be used in many surveys.

Word is often an adjective, which describes a certain Kansei. A word belongs to one or many languages.

Project helps a designer to organize and structure surveys. Each project can have many artifacts and many surveys.

Note is a short piece of text attached to a word. A designer, or subject in a survey, can scribble down an idea, question or other remark relating to the word, and save it in the system.

SurveyEntry is where the bulk of the collected data is kept. Every survey will yield a large number of entries, where subjects enter a value on how each artifact is related to a specific word.

Once these entities had been defined, they could be represented as tables in the database, as described in §4.2.3.

4.2.3 Database structure

Figure 4.1 displays a model of four objects whose relationships are very important for EvokeDB, and §4.2.2 added an additional six objects used to augment this model, in order to provide functions to automate surveys, analyze data and propose Kansei languages. However, EvokeDB is a web-application that relies on a database to store its objects, and therefore the database-structure in Table 4.2.3 was constructed with ten different tables along with attributes to represent the objects from §4.2.2.

Table 4.1: EvokeDB database structure

Artifacts	**Realms**	**Senses**	**Subjects**	**Surveys**
id	id	id	id	id
name	name	name	name	name
description	description	description	description	description
image_url	created_on	sensory_organ	date_of_birth	min_value
version	updated_on	created_on	gender	max_value
created_on		updated_on	nationality	language_id
updated_on			annual_income	project_id
			created_on	created_on
			updated_on	updated_on
Languages	**Words**	**Projects**	**Notes**	**Survey_entries**
id	id	id	id	id
name	name	name	author	attribute_id
description	description	description	content	subject_id
created_on	created_on	created_on	word_id	survey_id
updated_on	updated_on	updated_on	created_on	word_id
			updated_on	value
				created_on

This schema is needed for the functionality of EvokeDB, but it is by no means set in stone – new attributes can be added, thus giving a designer more control over the system and how it is configured. This type of flexibility is usually not easy to implement without re-writing parts of the application, but designers cannot be expected to perform such operations. Changes to attributes should be recognized and automatically handled by the application.

This problem presented me with a serious challenge. Up until this point I have described EvokeDB as a model with ten objects, implemented as tables with fields in a database. However, more advanced features, such as automated surveys, are not possible unless I can use these objects in a programming environment, and work with them using an object-oriented language. This requires a translation between the object-oriented layer and the database layer, which is capable of preserving the properties of the objects, as well as their relationships. Simple relationships between objects can be set up through a join-table containing only the foreign keys of the objects being joined, but this is difficult to maintain when relationships increase or become more complex, and it also requires complex SQL-queries to retrieve objects. Thus, an object-oriented programming language coupled to an efficient object-relational mapping for the database-tables was needed to bring functionality to EvokeDB. The Ruby-on-Rails framework [45] provided a solution, as described in §4.3.1.

4.2.4 On data retrieval and semantic search-algorithms

I will explain how user requests are handled by the system in the coming sections but I find it suitable to place EvokeDB within a context at this point. There has been a few other Kansei Database-systems presented prior to my research, where semantic associative search-algorithms are possible through careful definition of meta-data that describe contents (photos, recordings, videos) in a multimedia database, in terms of Kansei (e.g. [24, 25]). EvokeDB is not a multimedia database, although it contains some multimedia content used to present artifacts. Furthermore, it does not use meta-data to describe its contents; instead meaning is found through the rich set of relationships (see §4.3.1.2), which can be easily explored through its interface. This is a simpler approach, but one that fits well to the design intent of this system.

4.3 Application components

4.3.1 Ruby-on-Rails framework

Ruby is a modern object-oriented programming language created by Yukihiro Matsumot in the mid 1990's [44]. It has a very intuitive and clean syntax with similarities to both Perl and Python, and a design philosophy that focuses on ease-of-use for the programmer.

Rails [45] is younger yet - it was first developed in 2004 by David Heinemeier Hansson as a tool to automate the development of an on-line project-management application, and it has since then been extracted and released to the public. Active development has turned Rails into a very popular framework for building web-applications, due to the speed and ease of building complex applications with a database back-end.

There are many different programming languages one could use to make a Kansei database-system, but I advocate the Ruby-on-Rails approach as a very suitable candidate for this type of system, by reasons presented in the following sections.

4.3.1.1 Rails MVC-architecture

A Model-View-Controller (MVC) architecture gives an application structure by separating its data model, user interface, and control logic into three distinct components that can easily be manipulated and maintained by the programmer. In Rails, the *Model* consists of classes representing tables in a database. These classes inherit from the *ActiveRecord*-class, which provides an object-relational mapping and thereby methods to perform CRUD-operations (Create, Retrieve, Update and Delete). It is very easy to manipulate objects through this mapping, as Rails will handle communication with the database and relieve the programmer, or user of the system, of making complex SQL-queries. The *Model* will also contain code to describe the relationships between objects, whether they are *one-to-one*, *one-to-many*, or *many-to-many*. The *View* renders the *Model* into a form more suitable for interaction. It is used to display logic and communicate with the users. Each method in the *Controller* is connected to a *View*, which usually consists of a code-fragment in HTML (Hyper-Text Markup Language), with styling provided by CSS (Cascading Style-Sheet). Finally, the *Controller*-classes consist of code to handle user interaction and call the appropriate logic to respond to requests.

Ruby-on-Rails inherited object-relational mapping works without explicit configuration, as long as the tables in the database are named according to a specified convention of pluralism (for instance, object *Artifact* will map to table *artifacts*). "Convention over configuration" is a Rails mantra, and in most cases the constraints of following conventions are well worth the ease-of-use this philosophy brings. Code for many common tasks is automatically generated by Rails scaffolding-technique, which in essence creates a skeleton of an application by one simple command. This scaffold is not perfect by any means - it should be improved, replaced or augmented bit by bit, but it will still give a lot of functionality to an application in a very short amount of time and relieve the programmer of many tedious tasks.

4.3.1.2 Relationships in Rails

Section 4.2.3 described the problem of handling communication between objects and tables in the database in an efficient manner that would not require construction of complex SQL-queries. Luckily, Rails provides an object-relational mapping through the *ActiveRecord*-class, which can be used to set up relationships between objects with a few lines of code that will automate queries between the application and database. The class-definition in the model for a *Survey*-object is given below:

```
class Survey < ActiveRecord::Base
   belongs_to :language
   belongs_to :project
   has_and_belongs_to_many :senses
   has_many :survey_entries, :dependant => true
   has_many :artifacts, :through => :survey_entries
   has_many :words, :through => :survey_entries
   has_many :subjects, :through => :survey_entries
```

```
end
```

This declaration states that the *Survey* will inherit from the *ActiveRecord*-class and thereby obtain a number of methods to communicate with the underlying database.

Furthermore, there are a number of relationships that will automatically link to the appropriate tables in the database and add functionality to the model. For example, a survey belongs to a project, and a project can have many different surveys. A survey will be conducted with a specific language (set of words), but a language can also be used in many different surveys. These two relationships are called *one-to-many*, and are used frequently in EvokeDB. There are a number of *many-to-many* relationships as well, easily identified by the *has_and_belongs_to_many* and *has_many :through*-declarations. *ActiveRecord* has three basic types of relationships between tables, as described below.

One-to-one relationship A *one-to-one* relationship is created by using a foreign key in one row in one table to reference another row in another table, thus creating a link between the two. The corresponding models will have declarations of *belongs_to* and *has_one* inserted to make Rails aware of this relationship.

One-to-many relationship This relationship is declared by a pair of *belongs_to* and *has_many*-declarations entered into the respective models of the child- and parent-objects. Rails will be able to link to the right table through the foreign key (for *Surveys* this is language_id and project_id, see Table 4.2.3).

Many-to-many relationship For this type of relationship there are two approaches, each with its own merits and demerits. A simple relationship can be set up through a join table (*senses_surveys*) of foreign keys (sense_id, survey_id), as explained earlier in Section 4.2.3. Each model will have a declaration of *has_and_belongs_to_many* (habtm) to tell Rails about this relationship. With this type of relationship it is easy to link a survey to a set of senses, and likewise, a sense to a set of surveys. However, it is not possible to include more objects or give this link any attributes; it is and will never be more than a link between two objects. Relationships between multiple objects (N-way joins) and richer relationships where the link can carry attributes require the *has_many :through*-construct, where the join-table will actually be implemented as an object with its own attributes. The *survey_entries* is such a join-table. It is implemented as the *SurveyEntry*-object by Rails to link words, artifacts, subjects and surveys to a value (see Table 4.2.3). This table will contain the bulk of the data in EvokeDB, and the *SurveyEntry*-object will often be called upon to deliver data for an analysis. In the class-definition of *Survey* above we can see that the *survey_entries* has an option dependant declared as true. This ensures that if a survey, subject or artifact is deleted, then the corresponding *survey_entry* will also be deleted. Rails makes this type of database maintenance very easy to implement.

4.3.2 AJAX and on-line surveys

AJAX (Asynchronous JavaScript and XML) is yet another acronym that has received a lot of attention in the web-developing community during the last year. It is not a technology in itself, but rather an umbrella term representing a group of technologies and how they are used together.

XMLHttpRequest is one of the most important components of AJAX as it allows an object to exchange data asynchronously with a web-server through an independent connection channel between a web-applications' client-side and server-side [47]. This solves a serious drawback with web-applications, which up until now have left subjects in on-line, unsupervised surveys less supported in terms of how the system communicates with them. A traditional desktop application is very responsive - if a user clicks a menu or makes an input to the application, the result is usually displayed at once. However, the Web has been, and still is, a far more constrained medium in terms of how we interact with it. Basically, we are presented with content in a hypertext-system which we navigate through by clicking on links that tells the server to serve us new information; it is an interaction model of constant call-and-response that keeps users waiting and thus lacks the responsiveness of a desktop application. By implementing AJAX it is possible to create a user interface which communicates with the user in "real-time" without the constant need to submit data and reload the whole page to get a response back from the server.

For example, a common task in a survey to collect Kansei-data is to present a subject with an artifact, which is then evaluated by the semantic differential method [36] on a set of words (or rather, bipolar pairs of words) representing Kansei. This evaluation is usually performed by making a correlation of how strongly the artifact relates to the word on a range from low to high, either by entering a number, clicking on a radio-button, or adjusting the value on a slider. In a supervised survey a subject can always ask the interviewer about the input method or the meaning of the range, but this is not possible for large-scale, on-line surveys. It is therefore important to give clear instructions and explain the meaning of a range before the survey, as well as provide feedback during it. With AJAX, the system can respond to a users' input directly through a number of predefined events, and thereby provide support and make sure that the meaning of an input is properly understood. A traditional approach on the other hand, is bound by the limitations of the hypertext interaction model and can not cater for this type of feedback during an evaluation.

4.3.3 Apache web-server

Apache [3] is an open-source HTTP-server which EvokeDB will run on. It is available for a number of different platforms at apache.org. A number of other HTTP-servers, both commercial and open-source, are also available (Rails even comes with its own), so this choice would probably depend on the current IT-infrastructure of a company wishing to implement EvokeDB.

4.3.4 MySQL database

There are many databases that would fit the bill for this application and I am certainly not biased regarding which is the best one. I chose MySQL [29] since I am familiar with it and because it is offered under a GNU General Public License (GPL). Download available at dev.mysql.com.

4.3.5 PhpMyAdmin database administration tool

PhpMyAdmin [37] is a tool to handle administration of MySQL via an intuitive Web-interface. Again, it is not the only tool available for this task, and others may be just as suitable. For the more experienced user it is of course also possible to access the database directly through a command-line interface. However, a graphical user interface (GUI) will certainly be a helpful tool for the novice. Direct access to the database is mainly for management, addition of attributes to models, or retrieval of data through SQL-queries, but is in general not required for most projects once the database structure has been set up. PhpMyAdmin is available in 50 languages and offered under a GNU General Public License (GPL) at phpmyadmin.net.

4.3.6 System implementation

The number of components required might seem daunting at first, but they are in general easy to install and configure, and give a user a high level of control over the system. Figure 4.2 gives an overview of the components and the flow of requests between them. One positive aspect of this system is that it can run locally, or within the IT-infrastructure of a company, or exposed to the whole world through the Internet. This gives many options on how to conduct a survey.

The top of the figure displays two different interfaces - phpMyAdmin and EvokeDB. This is the client-side of the application. First of all there is a flow for direct access to the database through phpMyAdmin → Apache → MySQL. This gives a user access to administrate the database and configure EvokeDB, for example by changing attributes of objects. Secondly, there is the flow starting through the EvokeDB-interface. This flow will also communicate with the database but via the objects defined by Ruby-on-Rails. Any requests will go through the appropriate objects' model and controller, which in turn will query the database and present the user with an output (formatted by HTML, CSS, and AJAX) through a view. The interactions of a user (designer) and those of subjects taking a survey are of course not the same even though they use the same objects, but this is easily handled by defining different controllers for them.

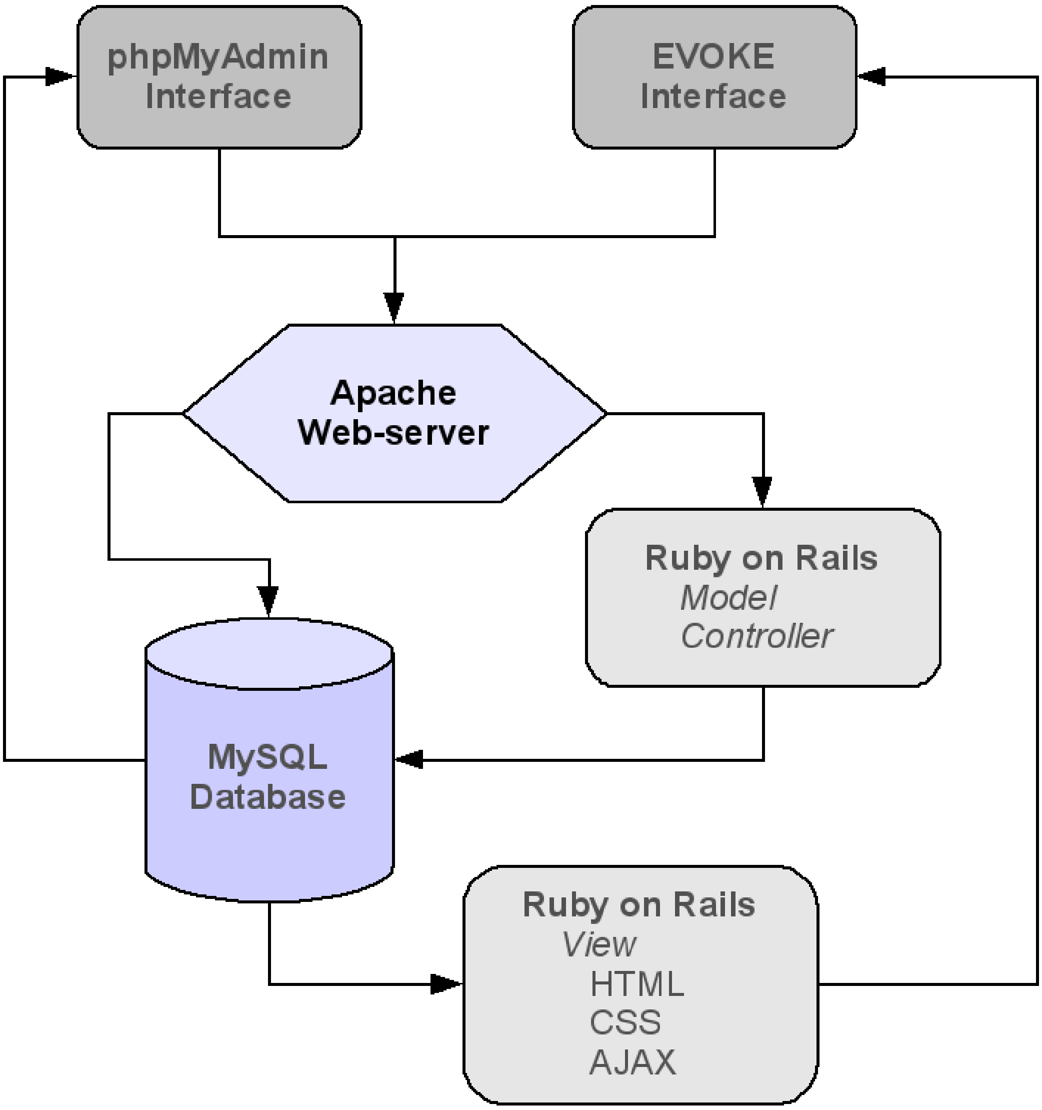

Figure 4.2: System components and flow

4.4 EvokeDB in action

In this section I will make a simulated run through the steps of the application to exemplify the flow and use of EvokeDB. Let us pretend we are an automaker in the premium segment, and that we want to explore Kansei of luxury cars in order to ensure the success of our upcoming model. Even as a top-tier company, we are on a limited budget and therefore we need to invest our resources wisely, where they count the most. However, we have an overwhelming amount of parameters to consider - should we put our effort into careful selection of the exterior shape, interior materials, colors, perform extensive sound engineering, or just simply put on a very nice set of rims? We need some guidance on how different stimuli will affect a customers feelings about our product, so we decide to make a survey to collect data. The seven steps in next section exemplifies the set-up of EvokeDB, and some considerations taken for this survey.

4.4.1 Perform initial configuration

For a new project there are a number of objects that we must create instances of, as described below.

Create a project The first task is to create a new project with a name, description and other attributes (owner, deliveries, deadlines?) we associate with our project. This helps to structure storage of data to allow us to reference it easily, even years down the line.

Enter artifacts Then we enter all the artifacts that belong to the project, and describe them properly. We select a number of cars in the luxury segment, and also include our new concepts. We also decide to use pictures of the cars as a means of presentation that will allow us to deploy this survey via our company IT-infrastructure. This is a limitation of the affective bandwidth, but we will validate our data as explained in chapter 3. The pictures are uploaded to the system so that EvokeDB can create a presentation of each artifact through a template that we can easily modify to fit our requirements on styling.

Define realms Now, here is our chance to structure the complexity of the car. We define four realms as *shape*, *color*, *brand_image* and *wheels*. The EvokeDB-interface also allows us to enter the stimuli of each realm, like *BMW*, *Mercedes Benz*, and *Bentley* for brand_image, and make all the links to the respective artifacts, simply by clicking on buttons.

Select words For this survey we are interested in Kansei in terms of luxury cars. As this is our first survey we do not have enough data from the past to allow EvokeDB to present us with a Kansei-language based on the realms we have entered. For now we just collect words from magazines, manuals and on-line resources relating to luxury cars and enter them into the database.

Group words into a Language We are likely to run a lot of surveys if this one turns out successful, so it would be good to create some structure by allowing words to be grouped in languages. We decide to create a language called "Luxury" and link it to a set of words: *luxurious*, *elegant*, *status*, *modern*, *expensive*, *comfortable*, *spacious* and *classic*.

Create survey We create a new survey and link it to the language and project we just created. A project can have many different surveys where each survey is about a specific language. Now we must also decide what type of range to use in the survey, and enter this into the min_value and max_value fields. We go with an AJAX-enabled slider between values 0 and 10 for this survey.

Link to senses As we only have pictures to represent the cars, our subjects are restricted to use their eyes to explore them. Therefore, we make a single link between our new survey and the sense of vision. This was our last task in the set-up of the survey.

Run survey The survey is now ready for deployment on our server. We decide to test it on our Intranet first, and invite a few colleagues by email. These people simply click on a link in the email to be transferred to our server where they are greeted and prompted to fill in their personal data. The

> *Subject*-object in EvokeDB will handle that form-entry and save the information in the database. Successful entry of a subject will call upon the *Survey*-object and request that our survey is launched. Without going into the specifics of any code, I will try to explain this in simple terms. Basically, what happens next is that the *Survey* asks *Project* how many *Artifacts* are included. In a similar fashion the *Survey* will also extract all the words in the *Language* it belongs to. Then, for each artifact the survey will create a web-page that gives an instruction (defined by us) and presents the artifact with pictures according to the design we defined in our template. A list of all the words are also included, with a slider to adjust the value of how strongly an artifact is associated with a particular word. The order of artifacts and words are randomized to avoid any statistical bias (it is also my experience that this helps combat fatigue in the subjects to some extent - a new order of words for each artifact forces them to focus instead of entering values by a pattern they easily adopt to when the list of words is static).

An example of a survey interface for a project such as the one I have outlined here is (partially) depicted in Figure 4.3 below.

Note how Kansei is evaluated - a slider is used to set a value in a range. This will allow subjects to enter their Kansei very precisely. The presentation of the word (i.e., "How elegant is this car?") and the text accompanying the slider can be defined by the designer of the survey. Furthermore, it is possible to change the range and even looks (size, color) of the slider, as it is simply defined in CSS. Initially the slider is set at zero and displays a message prompting the subject to enter his or her Kansei. As soon as the slider is adjusted it will give feedback to the subject by stating the current rating. This is one of the strengths with AJAX, it makes the application more responsive in terms of how information is displayed and communicated.

Another feature provided by AJAX can be seen in the dictionary. If a subject is uncertain of the meaning of a word, it is possible to simply click on it and a dictionary entry (or explanation provided by the designer) will emerge. This is a very subtle way to provide additional information without cluttering up the interface or confusing the subject by launching a new window like a traditional

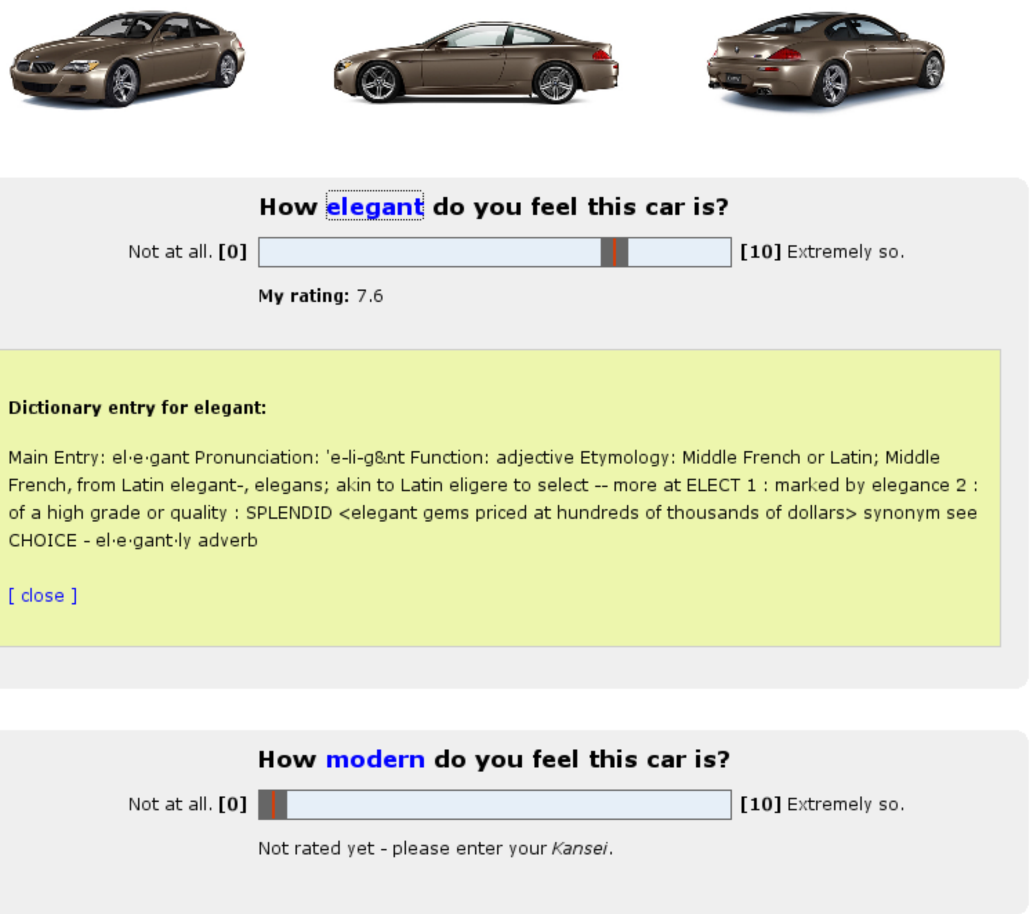

Figure 4.3: Example of survey interface

approach without AJAX would require. In case a subject wants to comment on a word, a similar feature of EvokeDB will allow this by providing a textbox. Such comments will create a new instance of a *Note* (see §4.2.2) and attach itself to the word.

Once all the words have been evaluated, the subject will click on a submit-button at the end of the form; this will send the data to *SurveyEntry*-object, which in turn will store it in the *survey_entries*-table in the database and request *Survey* to launch the next artifact. When there are no artifacts left to evaluate, the subject will be sent to a page where we thank him or her for the contribution to our survey. Again, this page can be chosen from a library of pages or created by templates.

4.4.2 Analyze results

Now that all the data is stored in our database, we can begin to analyze it to find some meaning within the Kansei it portrays. First of all, there are no real mysteries about this database, it can be accessed and queried like any other database, as depicted in Figure 4.2. It is possible to export its data or write SQL-queries from a command line or other interface to retrieve data for output. But, the new model and the relationships that come with it gives a number of possibilities to do low-level analysis of Kansei with predefined or automated SQL-queries, accessible through an easy-to-use interface of buttons and drop-down menus.

EvokeDB currently has some automation for this type of queries. Based on the relationships (between artifacts, realms, stimuli, and senses), the system can support a novice user with simple predefined queries. However, more specialized queries must still be entered manually. Still, this allows EvokeDB to function as a source of Kansei interpretation throughout a project - it can be referenced quickly to look up relationships without resorting to a full-fledged , time-consuming analysis. For example, with sufficient data it is possible to query the database not only like "What is the most elegant car?", but also use more subtle queries, like "How does black compare to red in terms of elegance?" or "How is *BMW* different from *Mercedes Benz*?" or "Are chromed rims more valued among men than women?". These relationships bring meaning to the collected data; meaning that has been obscured by simplified models. We still need a lot of data, but the relationships in EvokeDB are very easy to set up and therefore allows us to set up a more complex model of our data.

4.5 Discussion on Kansei Database Systems

The system presented here was spawned from what always griped me about Kansei Engineering - the lack of tools and guidelines, the "fuzziness" of how data is interpreted, the time and effort of making surveys, and the limited use of data. It was simply too much of an effort to implement. Data is the bread and butter of KE, and there is a need to simplify how we collect and store it. I envisioned a system that would automate surveys, and work as a sort of reference to Kansei - in short, it should be able to find the essence of various stimuli.

I quickly realized the power of associations, but also the difficulties of implementing them. Even in a face-to-face interview-situation it is difficult to make a subject "think-out loud" and capture a free chain of associations, and I was focusing on unsupervised surveys of large scale. However, this new model of mine, which links entities in KE by relationships, is a first step towards a deeper understanding of Kansei, and I hope to make further progress into this challenging area.

Currently, the system works very well for my initial design intention. It is easy to set up and launch surveys, and a user can store data and reference it easily through the interface. This gives more control, since there is a structure to all the entities involved in a project, and the system can offer some guidance in Kansei explorations. The analytical features, however, are still rather basic, and the predefined queries do not offer the best solution to efficiently retrieve and analyze data. I want to improve this before EvokeDB is released to a wider audience.

Many users are accustomed to information retrieval by keyword search, such as to those offered by search-engines on the Web, and I think this could be a feature that would bring a lot of usability to EvokeDB. However, a relational database is far more challenging to query than one consisting solely of text, due to its higher complexity in structure. In the last few years this problem has been addressed through active research, and there are today a number of systems available that can perform a keyword search on a relational database and rank the output efficiently (e.g. [1, 2, 4, 21, 28]). These results are promising for future development of EvokeDB, as an implementation of keyword search would present a user with the easiest possible interface for information retrieval.

My next area of concern is the definition of realms. As far as basic realms go, there is no real challenge to define them and share data between domains, but it soon becomes difficult to extract and categorize subtle stimuli from an artifact. Yet these small design cues, such as character lines in a car, can hold a great deal of the image portrayed by an artifact. Without them a Kansei evaluation may not be valid. The space of product properties can also be tricky to explicitly define (see Chapter 5 for an implicit method). However, Schütte [38] addressed these issues by collecting product properties from different sources, such as existing products, new concepts and company image. These properties were subjected to a selection process where three different groups (customers, experts, company) rated the importance to guide the final compilation step. This method requires additional work, but should provide a more robust set of chosen properties, and provide insight into possible realms.

4.6 Chapter Summary

Kansei Engineering (KE) is becoming more popular as a set of methods to create products and services that meet the emotional needs and wants from customers; an artifact should just feel right, and KE has the methods to succeed in this aspect. However, it is very difficult to implement KE in a design process without specialized training. In particular, surveys to collect data take time and skill to create and launch, with few specialized tools available.

Furthermore, there is a lack of guidelines on how to extract design parameters from an artifact - complex artifacts can lead to many different associations in a person, and this can in turn influence how the artifact is perceived emotionally. Because of this, it is hard to determine what property or characteristic of an artifact is most strongly correlated to a certain Kansei. Simplified models do not consider how an artifact is explored, or the context it is presented within. The study presented in this chapter introduced a new model of Kansei, in order to provide a foundation for a database-system with rich relationships. This system is called EvokeDB (Evaluation and Verification of Objects in Kansei Engineering via Database), and it uses relationships between words, artifacts, realms, senses and subjects, to provide a more in-depth exploration of Kansei.

This chapter outlined the foundation and technologies used to build EvokeDB. A database-structure, which is based on the new model, was presented. The Ruby-on-Rails approach is explained, as it provided an elegant solution to the challenge of mapping objects and their relationships to tables in the database. AJAX was presented as a set of methods to bring usability and responsiveness to a web-application, and its merits for gathering Kansei-data was highlighted. Finally, the set-up of a simulated project was explained, along with a description of survey-automation and the graphical user interface.

This system shows that on-line surveys can be automated and thereby become accessible even for users with limited knowledge in computer programming. EvokeDB can offer designers and product developers a tool to gather, store and analyze Kansei data with less need for specialized training. A tool like this can easily be implemented by open-source components, and used throughout a design process.

Chapter 5

Implicit Shape Parameterization

Automotive design involves very complex surfaces that are constructed under rigorous constraints; a shape must not only appeal to a customer from an aesthetic point of view, but it must also give the car good aerodynamics or even serve as a structural element. Perhaps most important of all, is that it can be mass-produced and easily handled in assembly for a low cost. The automotive industry has driven the development of information systems to support these processes, and there are an abundance of systems available on the market today; CAD, CAM, CAE, CFD and PLM are just a part of the alphabet-soup of technologies that provide very advanced tools for detail design, analysis and production. There is, however, one type of support systems that still remains relatively uncharted, and that is the analysis of emotional content (Kansei), in products.

Kansei Design and Engineering have become a very important part of developing products that stand out; since mass-production and fierce competition have brought customers products of similar level of functionality and quality, they are now in many cases quantitatively satisfied and will choose products that appeal to them on an emotional level. On-going research in Kansei has resulted in a number of methods to support design decisions by measuring emotional impact of products, and building models to predict responses of customers. The main objective in the Kansei field is ultimately to construct systems that are more sensible and responsive to human emotion.

Sadly, Kansei methodologies are still challenging to implement correctly. Extracting and transforming emotions into parameters suitable for analysis is not a trivial task; many sources of noise or error in the process can easily yield misleading results. Designers are usually not trained in data analysis, and the tasks of a typical Kansei study do not correspond well with a designer's skill set. It is a real paradox that we, as researchers in Kansei methodologies, strive to create methods that will map emotions and bring understanding to customer desires, but still the tools we provide are so poorly adapted to our own customers, the designers and product developers. This chapter presents a method that has a rather heavy, although robust, theoretical foundation, but the aim is that this theory shall be embedded into a CAD-application where designers can work in their comfort zone, with tools they are already accustomed to. In essence, I would like to bring Kansei design closer to the designer, and allow for a very intuitive exploration of design spaces based on 3D-modeling.

My discussion throughout this thesis has been restricted to shape design and in particular to shapes in the automotive domain, but I believe my results are applicable to any product with surfaces in 3D. Furthermore, in this chapter I use the terms "shape", "concept", and "surface" rather loosely depending on the context, but it should in all cases be clear that I refer to the artifact under study, in this case the front bumper used as an example (see Figure 5.1).

5.1 Parameterizing design features

A typical task in a Kansei study is to define parameters of a product, which are consequently used as vectors to span a design space. This is currently done explicitly without any clear guidelines, and I argue that this approach is unsuitable for exploratory analysis of complex shapes. I will refer to this as "explicit parameterization" of shape design features.

Explicit parameterization is a "common sense"-approach where a product is examined, usually visually, to identify the obvious main design features. The features are then represented by a parameterization so that an analysis with mathematical tools can take place. This approach therefore requires two error-prone steps to transform a shape into a set of numbers - that is, a designer must first know which features are important, and secondly, how to describe them numerically without loss of the emotional content the product contains. Naturally, it is very difficult to identify which particular features are most important with any degree of confidence.

Designers have a highly developed sense of shape, but these skills are expressed through active work with surfaces, where shapes are explored by drawing, sculpting and modeling to find the desired shape by *feeling*. The selection of a numerical representation for shapes by a designer is therefore an unnatural process that is likely to yield highly biased results, which may not accurately present the emotional content. I believe that we need a more intuitive approach to feature selection, one that is closer to the designer's real expertise. Another important aspect of a good parameterization is the number of extracted parameters, or the problem dimensionality, which will have a significant impact on the workload and complexity involved with collecting data and building models, as explained in §2.4.

In my previous work [34], I have used the silhouette lines of cars as a low-dimensional descriptor of their unique character, but I came to realize that modern cars exhibit very small deviations in this descriptor as they are all subjected to the same spatial and aerodynamic constraints. The global shape of modern cars does not have a large degree of variance between cars of the same type (compact, coupé, sedan, etc.), instead the brand identity is expressed as small deviations from a mean, with characteristic proportions and lines defining details in the shape (this very idea is the starting-point for the method I will describe in this chapter). For example, wide wheel-arches or an aggressive front bumper can signify a more "sporty" car. Any descriptor that does not consider all these proportions, lines and details, will not be able to sufficiently express the car's characteristics.

This poses a challenging problem; it is possible to describe the shape of a car in all its minute details, but this will lead to a design space with thousands of dimensions. A high-dimensional space, on the other hand, would not allow for efficient measurement of Kansei in a survey, since each vector or parameter must be accounted for, even if the number of required runs (or design concepts) are reduce by using the DOE described in §2.2.2. It would be very time-consuming to extract and parameterize all those subtle details in a design, thus rendering this approach useless in practical applications.

Therefore an accurate and low dimensional descriptor for complex shapes is needed. This descriptor should, by my earlier arguments, be defined by intrinsic properties of surfaces, and not by a designer's guesswork. In this chapter I will present a method to define such a descriptor by performing Principal Component Analysis (PCA) on point-clouds from an ensemble of 3D-shapes (point-clouds) spanning a design space. My contributions boil down to two significant points:

1. Implicit parameterization of 3D design features with an accurate and low-dimensional descriptor.
2. Intuitive definition of design spaces. A designer can load design features by using concepts modeled in any CAD-program capable of handling meshes or point-clouds.

5.2 Theoretical background

Principal Component Analysis (also known as Eigenmode Analysis, Karhunen-Loeve Expansion) is a popular procedure for extracting a basis for a modal decomposition from an ensemble of signals, such as the shapes presented in this study, due to its efficient reduction of a high-dimensional process unto a few dominant modes. In this section I will briefly explain the fundamentals of PCA (as it applies to this method), and also describe some important concepts of the snap-shot method [40], which enables modal decomposition of huge datasets.

Consider an ensemble $\{u^{(k)}\}$ of scalar fields, where each scalar field is a function $u^{(k)}(x)$ where x is in $1, \ldots, N$. Assuming that these functions belong to a Hilbert space $\mathbb{L}^2$, it is possible to find a basis $\{\varphi_n\}$ for $\mathbb{L}^2$ such that projections of members in the ensemble onto this basis yields a reduced, yet accurate representation of them. In short, we are searching for the optimal basis that will account for the maximum variance in the distribution of data in $u^{(k)}(x)$. This basis is defined as

$$\hat{u}(x,t) = \sum a_n(t)\varphi_n(x) + \bar{u}(x) \tag{5.1}$$

with a deviation from mean

$$v(x,t) = u(x,t) - \bar{u} \tag{5.2}$$

The orthonormal basis functions $\{\varphi_n(x)\}$ can be calculated (see [41] for details) via the set of linear equations

$$\int R(x,y)\varphi_n(y)dy = \lambda_n \varphi_n(x) \tag{5.3}$$

where λ_n is the eigenvalue and the covariance matrix is given by

$$R(x,x') = \frac{1}{N}\sum_{t=1}^{N} v(x)v^*(x') \tag{5.4}$$

Therefore, the optimal basis is given by the eigenfunctions φ_n of Equation (5.3), whose kernel is the averaged autocorrelation function. Choose λ_n such that

$$\lambda_n = \frac{1}{N}\sum_{n=1}^{N} |\langle \varphi_n, v\rangle|^2 \tag{5.5}$$

This can be interpreted as a mean energy projection. Therefore, the eigenvalues with corresponding eigenfunctions describe the mean energy of the process $v(x,t)$ on the φ_n-axis in function space. It can be shown that this basis will on average contain the most possible energy compared to all other linear combinations, thus making it the optimal one.

From Equation (5.4) it is evident that the calculation of the covariance matrix will become very costly, if not intractable, for a high-dimensional v. The method presented in this chapter uses surfaces in a discretized spatial domain, which allows for a high resolution but as a consequence also requires a high-dimensional v to accurately represent variations in the surfaces. These surfaces belong to a linearly independent set of M snapshot data samples, which can be seen as characteristics of the designers intent, or the signal the designer wants to transmit.

Now, if we assume that $u(x,t)$ is an ergodic process, then it is possible to represent the averaged spatial correlation function as:

$$R(x,y) = \lim_{T\to\infty} \frac{1}{T}\int_0^T v(x,t)v(y,t)dt \tag{5.6}$$

This reduces the problem of finding eigenvectors spanning an optimal basis to a far more tractable problem of finding the eigenvectors f_n and eigenvalues λ_n of the MxM matrix $\mathbf{C}$ given by

$$\mathbf{C}_{ij} = \frac{1}{M} \int v^{(i)}(x) v^{(j)}(x) dx \tag{5.7}$$

where $i, j = 1, 2, \ldots, M$. The empirical eigenfunctions are computed as linear combinations of the data snap-shots (i.e., the concepts in the ensemble, see §5.3.1) via:

$$\varphi_n(x) = [v^{(1)}\, v^{(2)}\, \cdots\, v^{(M)}] f^n \tag{5.8}$$

where $n = 1, 2, \cdots, M$. This means that the snap-shot method reduces the decomposition from a problem of order $\mathbf{O}(N^2)$ to a far less costly calculation of order $O(M^2)$. Full details on the snap-shot method and the analysis of large datasets are presented in [40].

5.3 Method

The steps required to create a low-dimensional basis by PCA are outlined below:

1. Load an ensemble of design concepts (3D-mesh models) to span design features.
2. Place a uniformly distributed point-cloud on each concept.
3. Find one-to-one point-correspondences for the point-clouds within the ensemble.
4. Calculate eigenshapes as low-dimensional shape descriptor from dataset of aligned point-clouds.

In some cases where the meshes for design concepts already have ordered, close to equilateral triangles, it may be possible to skip step 2 and 3, and use the vertex positions directly.

5.3.1 Concepts

Concepts are 3D-models that belong to an ensemble of shapes, which in essence will build the space of possible shapes. There are basically two different ways to obtain concepts – they are either created exclusively for a study, or existing ones are modified and used. I have used both these methods to create samples for several studies. The modeling was performed in both Alias Studiotools^{TM}(a commercial software-package used by many designers in the automotive field), as well as in Blender^{TM}(open-source under GNU/GPL-license).

A designer or modeler is very accustomed to working with CAD-tools, and he or she can therefore freely express various design ideas as concepts. Most CAD-programs available on the market today have functions to convert or save models as a mesh-representation even if the modeling-technique itself is based on some other representation. This allows designers to remain in their field of expertise, where their input can be maximized.

An important requirement for these concepts is that they should all belong to the same family of surfaces and share topology. For example, inclusion of a cabriolet (open car) in an ensemble of sedans is likely to yield poor results, because the point-correspondence algorithm in §5.3.3 will not be able to match points on these fundamentally different surfaces. This level of similarity is easy to follow when concepts are constructed from scratch. An easy approach is to start from a given subdivided surface and only make small adjustments to control-points to define design features for an ensemble of concepts, such as the one depicted in Figure 5.1.

Mesh-models are common in the computer graphics industry due to their efficient representation of 3D-objects, but the automotive industry has also seen an increased use of meshes in some steps of the design process, because the measuring and data transfer of real objects to 3D-representations have become standard procedures. It is therefore vital that this huge pool of previous work can be utilized. Existing models, from past work or the extensive resources that are available in both commercial and free libraries on-line, can be used as concepts with less need for time-consuming 3D-modeling. However, in most cases this approach still requires some pre-processing of the meshes to make sure that the selected surfaces are indeed similar, so that a point-correspondence can be found between all concepts. Section 5.3.2 and 5.3.3 will give an overview of the preprocessing used in this case.

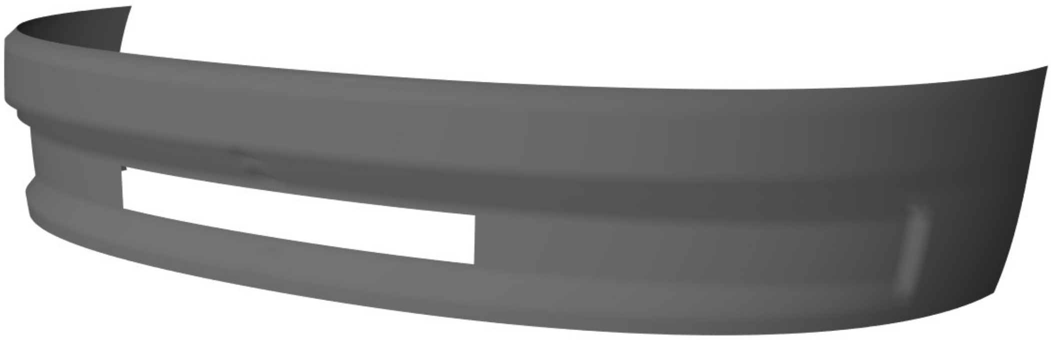

Figure 5.1: Front bumper

5.3.2 Point-clouds

The meshes used as concepts are in general not suitable for analysis without further preprocessing, as they exhibit a great deal of variance in the distribution of vertices. A nonuniform distribution of unordered vertices can result from procedures to create more economical representations of surfaces, where surface-areas with a high degree of variance will employ a higher density of points, and thus the faces that makes up the mesh will come in a variety of sizes and shapes. However, this presents a problem since this data cannot easily be used to compare different shapes - we need to create point-clouds on these surfaces, which shall have a uniform distribution with close to equal distances between points. This will provide a much better starting-point for the point-correspondence algorithm of §5.3.3. The following algorithm was used to place N uniformly distributed points on a mesh-surface:

1. Calculate the probability P of a point landing on a specific face as the area of that face compared to the total area of the surface (i.e., relative area).

2. Select a random face weighted by P, and place the point randomly on this face. Repeat N times for a randomly distributed point-cloud.

3. Apply a repulsion forcefield based on point proximity to re-distribute the points over the surface. Repeat until equilibrium is reached.

An area-weighted probability is necessary to create a random distribution of points on meshes which may have faces of various size. This random distribution must undergo a relaxation process, where points are moved around until they are spaced regularly. Turk [46] presented an outline of this process as:

```
loop k times
  for each point P on surface
    determine nearby points to P
    map these points onto the plane containing the face of P
    compute the resulting repulsion force F that mapped points exert on P
  for each point P on surface
    compute new position of P based on the repulsion force F
```

Figure 5.2 shows the surface for a front bumper in two different representations (note that a bumper is usually symmetric, so only one half is modeled and then mirrored to obtain the full part).

The repulsive forcefield used in this study was based on a radius of repulsion:

$$r = 2\sqrt{\frac{A}{\pi N}} \tag{5.9}$$

where A is the total area of the mesh and N is the number of points. Points within this radius will exert a force that falls off linearly with distance and closely clustered points will push each other away to a more stable state. During my tests I found that this force should be controlled so that points will only move small distances on each iteration; if points are pushed too far it is likely that they will enter another cluster and disturb fast convergence to a stable state.

The initial random placement of points will often give very tight clusters of points, and other less populated areas. This depends on the “randomness” of the algorithm used, but running the relaxing algorithm in a few sets with a different

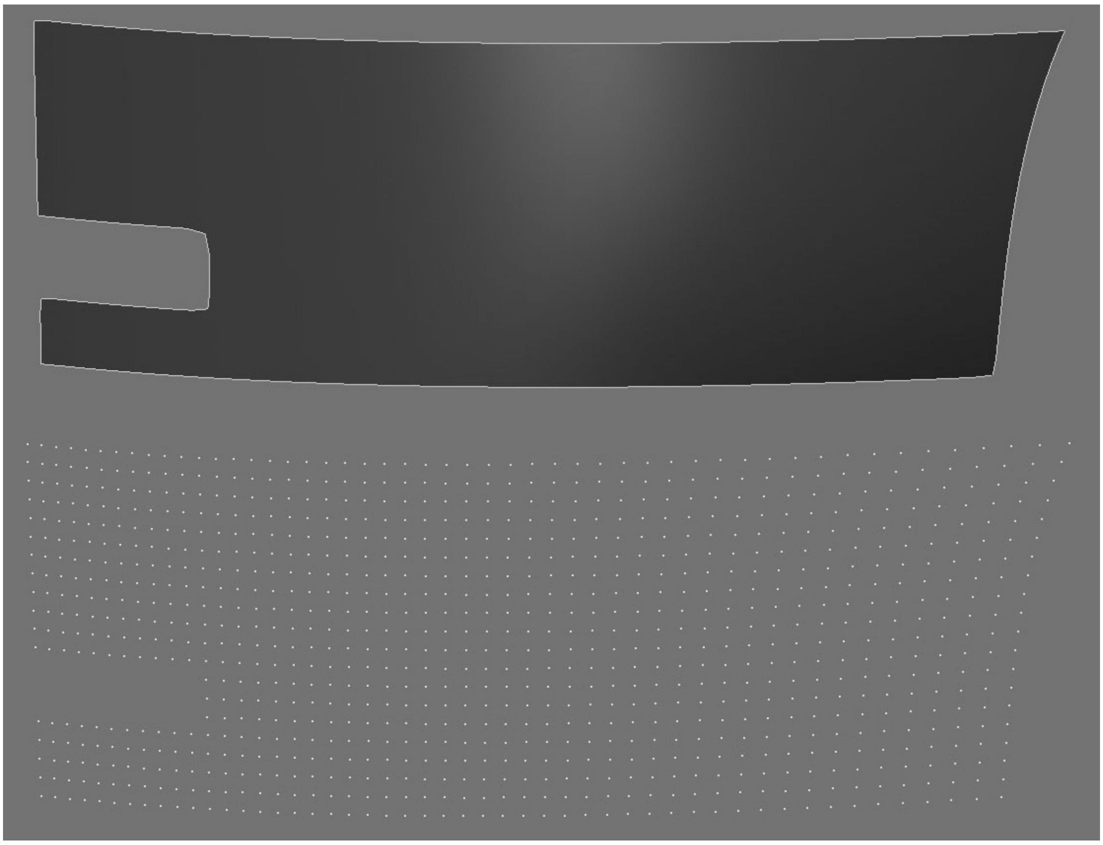

Figure 5.2: Surface as mesh and point-cloud

radius of repulsion r improved the distribution and convergence - I typically start with a smaller r to safely break up tight clusters before I use the r from Equation (5.9). A few iterations with a larger r will avoid local minima by forcing points, based on a larger neighborhood, into less populated areas. It is also important to make sure that the points are not pushed away from the surface, so each iteration will carefully monitor the new position, check if an edge was crossed, relocate the point to a neighboring face if needed, or move on the boundary edge of the surface.

This process of repulsion works well for smooth surfaces, but will yield poor results for folded surfaces (or surfaces with very sharp sections) as it may falsely report points as neighbors in these cases. There is also a level of resolution to consider - the point-cloud has no reduction built in, and therefore very fine details in a surface can only be encoded by a brute-force approach which translates into a very dense points-cloud, with thousands of points.

5.3.3 Point-correspondence

The random placement of points, and the relaxation process that followed produced a point-cloud for each concept in the ensemble. However, these point-clouds are unordered, and must be aligned so that each point corresponds to its equivalent points in the rest of the point-clouds. This is a complex problem, especially for concepts with a large degree of variance or very dense point-clouds.

A number of schemes ([22, 39, 48]) have been presented to calculate point-correspondence for 3D-shapes, and most use the spectral domain to obtain invariance to rigid transformations, scaling and rotation. I used the algorithm proposed by Jain et al. [22] to compute a one-to-one point-correspondence between a selected concept with all remaining concepts in the ensemble, and I will only give a brief overview of the matching below:

1. Calculate an affinity matrix $\mathbf{A} = a_{ij}$ for the first point-cloud, and likewise $\mathbf{B} = b_{ij}$ for the second. These are symmetric matrices, where a_{ij} and b_{ij} are the euclidean distances between point i and j in each cloud.

2. Find the k-dimensional spectral embeddings $\mathbf{A}_k$ and $\mathbf{B}_k$, which are based on the eigenvectors of $\mathbf{A}$ and $\mathbf{B}$.

3. Align the spectral decompositions iteratively, by a greedy or exhaustive search.

4. Compute a correspondence by minimizing a cost-function based on euclidean distance between the spectral-decomposed point affinities. This procedure is repeated for all concepts in the set to obtain a fully aligned ensemble of uniform point-clouds, which will be used to calculate eigenshapes.

5.3.4 Eigenshapes

Consider the ensemble of point-clouds, where each concept is a point in a huge space (if we place 1000 points on each concept, we can think of this as a system with 3000 degrees of freedom, since each point is described with three parameters in Euclidean space). However, since the concepts have a similarity (they are all front bumpers, not bicycles or rice-cookers), it is easy to understand that the ensemble will occupy a small part of that space, that is, they will be clustered together. Therefore, it is possible to describe them as a much smaller subset of the original space.

PCA was used, as described in §5.2, to extract successive vectors which accounted for the maximum variability in the distribution of points. These vectors, which I would like to refer to as "eigenshapes" (similar to "eigenpictures" in [41]), span the subspace of the initial design concepts, and can therefore be considered as descriptors of the designers intent. Each eigenshape is a linear combination of the original shapes, and encodes unique features. The necessary steps for finding eigenshapes from an ensemble of M concepts (point-clouds) are:

1. Flatten each point-cloud into a 1D-vector $u(x)$. The ensemble is $u(x,t)$, where t is the index of each concept.

2. Calculate the mean shape as $\sum u(x,t)$ divided by the number of concepts.

3. Calculate the deviation from mean shape from Equation (5.2) and save the result as vectors $v^{(m)}(x,t)$. These vectors can be interpreted as the unique design features of each shape.

4. Find eigenvalues λ_n and eigenvectors $\mathbf{f}_n$ from matrix $\mathbf{C}$ in Equation (5.7).

5. Calculate eigenshapes φ_n from Equation (5.8).

From Equation (5.1) we now have a low-dimensional basis that can be used as a shape descriptor in Kansei Engineering studies, where efficient Kansei measurement, analysis, and model construction is only practically feasible for a low number of parameters.

5.4 Results

Consider a study of front bumpers like the one in Figure 5.1. I have created a set of concepts with various design features, of which a selection is shown below. These concepts all follow spatial constraints, such as the fitting to the wheel, but there are also different design features introduced, which will be encoded into this design space. Each concept in this ensemble can be expressed as:

$$u \approx \sum_{n}^{M} a_n \varphi_n + \bar{u} = u^s \tag{5.10}$$

where s indicates the number of eigenvectors used in the linear combination.

To get a qualitative measure of how close this reconstructed shape is to the original shape, a useful measure of error is given by:

$$E_s = \frac{\|u - u^M\|}{\|u\|}. \tag{5.11}$$

Figure 5.4 shows a plot of $100E_s$ for nine different concepts, along with average error. The error is very small, and decreasing as more eigenvectors are used. I used control-points of a given subdivided surface to create concepts with well-behaved surfaces for this study, and the small error follows from this construction method.

Figure 5.3: Ensemble of design concepts for front bumber

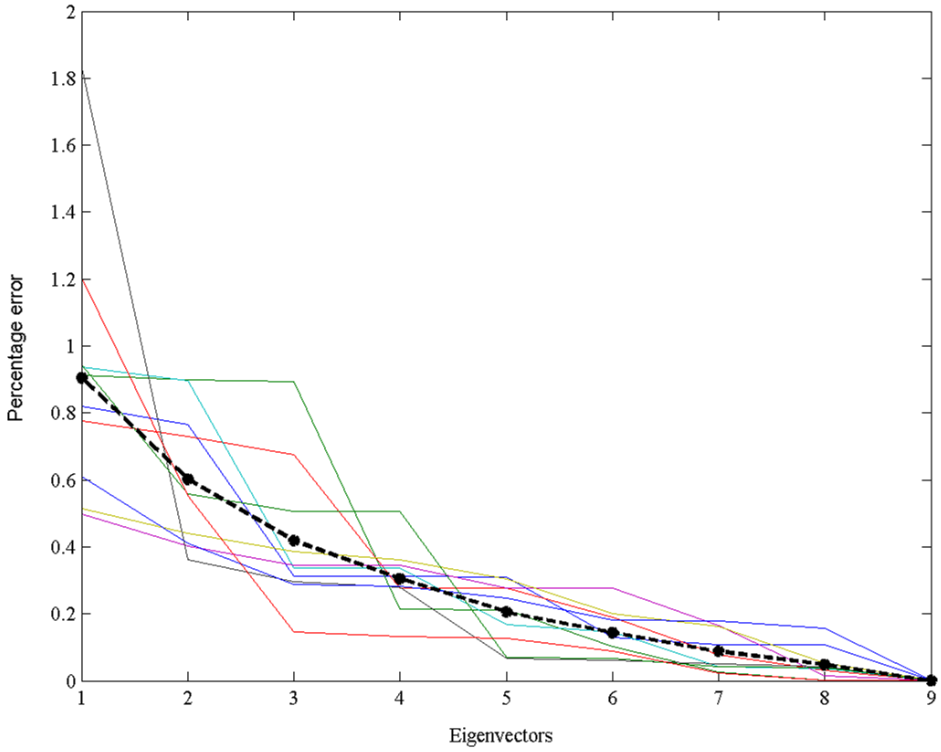

Figure 5.4: Percentage error, shapes reconstructed from Eigenshapes

Section 5.3.2 and 5.3.3 presents techniques to generate point-clouds from existing meshes, and from my experiments with that approach I have in general had a higher level of error, but still with the same characteristics of the spectrum shown in Figure 5.5. This plot of the eigenvalue spectrum tells a similar story - only a few eigenvectors account for a large amount of variance in the dataset. If only three eigenvectors are retained, more than 90% of the variance will still be accounted for, which ensures accuracy. That gives only three parameters to use in the analysis of a rather complex surface, which previously has been very difficult to parameterize for a Kansei study.

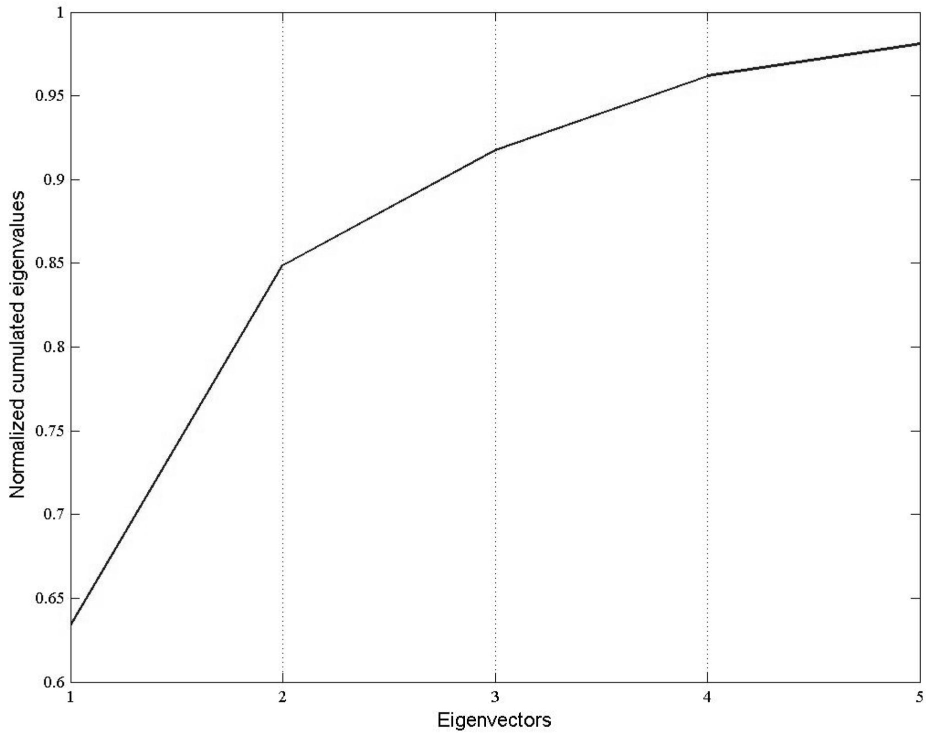

Figure 5.5: Eigenvalue spectrum

The shape space constructed in this manner will respect boundary points, and therefore the shape generating algorithm in [34], which is working within this space, will only create shapes that are within spatial constraints. The limitations of the space lies in the concepts, they are effectively defining the features that can be explored. Since every new generated surface is a linear combination of eigenshapes, and therefore stems from the concepts, there is no way to obtain “new” features.

5.5 Chapter summary

The method present in this paper solves two major problems in applying Kansei Engineering to shape design.

First, it gives an implicit parameterization of shape design features, thereby relieving the designer of selecting and parameterizing design features that could yield misleading results. This way, it allows a designer to explore shapes and work with Kansei methods in a very intuitive and familiar way.

Secondly, it provides a low-dimensional basis for complex shapes while still encoding small design features and respecting constraints. It is therefore both accurate and economical. The method reduces a source of error that is related to the designer bias to obvious, visual design features that may or may not be important for the emotional content within an artifact. It also brings a lot of practicality to the analysis of complex shapes that have been difficult to analyze and gather data on, because there are too many design parameters involved.

The drawback lies in the requirement for a time-consuming creation of the concepts that define the design space, but this task can also be reduced by using existing 3D-data that has been processed to form uniform point-clouds with point-correspondence. The data-processing also presents a heavy load, but this work can be automated and does not require human intervention, which makes this method suitable for implementation in CAD-tools. Shape exploration and Kansei studies of products with complex shapes are certainly possible to perform in an accurate and economical manner by using the procedure I have laid out in this chapter.

Chapter 6

Conclusions

The first production Model T Ford was assembled at a Detroit plant in 1908. It became immensely popular as a car for the masses during the next 19 years of production. New innovative manufacturing methods made this car possible, and Henry Ford is said to have made the statement "Any customer can have a car painted any color that he wants so long as it is black". True or not, this story is still very interesting when compared to the situation in the automotive industry today – customers can now choose a car to their liking based on very personal tastes, preferences, and more than ever, *desires*.

Technology can bring new functionality and improve performance, but in many cases customers are already quantitatively satisfied, and will choose a car by the feelings it evokes in them. The car should "feel" right, and match their desires in an intuitive way that is very difficult to measure and understand with traditional methods in product development. Kansei Engineering provides tools to explore the emotional aspects of products, and although frameworks and methodologies have been presented, there is still a lack of a clearly defined nomenclature, and systematic guidelines on how to perform a study on shape design. Chapter 1 contains some definitions on Kansei, as well as the aim of this study and how it relates to previous work.

Furthermore, I have showed that a design support system with shape sensibility is possible to build by following the Kansei Design Methodology in Chapter 2. The Kansei Design process still requires a fair amount of work and analytical thinking from its practitioner, but I hope I have at least lowered the threshold and provided an insight into the process and some of it parts, so that interested parties can get their own systems up and running.

In Chapter 3 I have pointed out some dangers of using representations of real artifacts to collect Kansei data. Such surveys should validate the accuracy of a representation and match Kansei words, and build models accordingly. It is very easy to get an answer from a model, but it is far more difficult to have confidence in it. Surveys are in general very time-consuming to set up, but with the database and framework I have presented in Chapter 4 I believe that large-scale, on-line surveys can be performed by non-experts with very little introduction to the system.

Finally, although my algorithm for implicit shape parameterization in Chapter 5 is theoretically and computationally heavy, I feel that this is the most important contribution in my thesis. This method has the potential to bring Kansei design algorithms to commercial CAD-systems – explicit shape parameterization requires custom-made shape generators, but my implicit approach allows any CAD-system capable of handling point-clouds and meshes to implement algorithms for analysis of emotion in shapes. Hopefully, this work will inspire designers and system developers to create systems that support more intuitive exploration of shapes; one that considers all the emotional aspects of a product, and the associations and emotions it evokes from its human users.

References

[1] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: A system for keyword-based search over relational databases. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, pages 5–16, 2002.

[2] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: enabling keyword search over relational databases. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 627–627, 2002.

[3] Apache. Company website, 2007. URL `http://www.apache.org`.

[4] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using banks. In *Proceedings of the 18th International Conference on Data Engineering*, pages 431–440, 2002.

[5] G.E.P. Box, W.G. Hunter, and J.S. Hunter. *Statistics for Experimenters.* Wiley-Interscience, New York, NY, USA, 1978.

[6] R.B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276, 1966.

[7] L. Chen, G. F. Wang, K. Hsiao, and J. Liang. Affective product shapes through image morphing. In *DPPI '03: Proceedings of the 2003 international conference on Designing pleasurable products and interfaces*, pages 11–16, 2003.

[8] M. Creusen and J. Schoormans. The different roles of product appearance in consumer choice*. *Journal of Product Innovation Management*, 22(1):63–81, 2005.

[9] V. Czitrom. One-factor-at-a-time versus designed experiments. *The American Statistician*, 53(2):126–131, 1999.

[10] P.M.A. Desmet. *Designing Emotions*. PhD thesis, Delft University of Technology, 2002.

[11] P.M.A. Desmet, C.J. Overbeeke, and S. Tax. Designing products with added emotional value: development and application of an approach for research through design. *The Design Journal*, 4(1):32–47, 2001.

[12] G.E. Farin. *Curves and surfaces for computer aided geometric design: A practical guide*. Academic Press, San Diego, CA, USA, fourth edition, 1997.

[13] J.A. Ferwerda, S.H. Westin, R. C. Smith, and R. Pawlicki. Effects of rendering on shape perception in automobile design. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 107–114, 2004.

[14] J.H. Friedman. *An overview of prediction learning and function approximation; From Statistics to Neural Networks: Theory and Pattern Recognition Applications*. Springer-Verlag, New York, NY, USA, 1995.

[15] R. Gorsuch. *Factor Analysis*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, USA, second edition, 1983.

[16] J.F. Hair, R.E. Anderson, R.L. Tatham, and W.C. Black. *Multivariate Data Analysis*. Prentice Hall, Upper Saddle River, NJ, USA, fifth edition, 1998.

[17] A. Harada. The framework of kansei engineering. *Report of Modelling the Evaluation Structure of Kansei*, pages 49–55, 1997.

[18] A. Harada. Modelling the evaluation structure of kansei using network robot. *Report of Modelling the Evaluation Structure of Kansei II*, pages 15–19, 1998.

[19] H.H. Harman. *Modern Factor Analysis.* University of Chicago Press, Chicago, Il, USA, third edition, 1976.

[20] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice-Hall, Upper Saddle River, NJ, USA, second edition, 1999.

[21] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient ir-style keyword search over relational databases. In *Proceedings of the 29th VLDB Conference*, pages 1–12, 2003.

[22] V. Jain and H. Zhang. Robust 3d shape correspondence in the spectral domain. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, page 19, 2006.

[23] H.F. Kaiser. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20(1):141–151, 1960.

[24] T. Kitagawa and Y. Kiyoko. A mathematical model of meaning and its application to multidatabase systems. In *Proceedings of the 3rd IEEE International Workshop on Research Issues on Data Engineering; Interoperability in Multidatabase Systems*, pages 130–135, 1993.

[25] Y. Kiyoki, T. Kitagawa, and T. Hayama. A metadatabase system for semantic image search by a mathematical model of meaning. *SIGMOD Rec.*, 23(4):34–41, 1994.

[26] Kyosho. Company website, 2007. URL `http://www.kyosho.co.jp`.

[27] S.H. Lee, A. Harada, and P.J. Stappers. *Pleasure with products: design based on Kansei, In: W.S. Green, and P.J. Jordan, Pleasure with Products: beyond usability.* Taylor & Francis, London, UK, 1999.

[28] F. Liu, C. Yu, W. Meng, and A. Chowdhury. Effective keyword search in relational databases. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 563–574, 2006.

[29] MySQL. Company website, 2007. URL `http://www.mysql.org`.

[30] M. Nagamachi. Kansei engineering: A new ergonomic consumer-oriented technology for product development. *International Journal of Industrial Ergonomics*, 15(1):3–11, 1995.

[31] M. Nagamachi. *Kansei Engineering: The Framework and Methods*. Kaibundo Publishing Co., Ltd., Kure, Japan, 1997.

[32] S. Nagasawa. Present state of kansei engineeering in japan. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 333–338, 2004.

[33] B. Nixon. Evaluating design performance. *International Journal of Technology Management*, 17(7-8):814–829, 1999.

[34] A. Nordgren and H. Aoyama. Style design method based on form impressions. In *Proceedings of IDETC/CIE 2005, 31st Design Automation Conference*, pages 1–8, 2005.

[35] A. Nordgren and H. Aoyama. Evokedb – a kansei database system for automated surveys. *Kansei Engineering International*, 7(1):1–10, 2007.

[36] C.E. Osgood, G.J. Suci, and P.H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, Champaign, Il, USA, 1957.

[37] PhpMyAdmin. Company website, 2007. URL `http://www.phpmyadmin.net`.

[38] S. Schütte. Developing the space of product properties supporting kansei engineering procedure. *International Journal of Kansei Engineering*, 5(2): 11–19, 2006.

[39] L.S. Shapiro and J.M. Brady. Feature based correspondence: An eigenvector approach. *Image and Vision Computing*, 10(5):283–288, 1992.

[40] L. Sirovich and R. Everson. Management and analysis of large scientific datasets. *The International Journal of Supercomputer Applications*, 6(1): 50–68, 1992.

[41] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America*, 4(3):519–524, 1987.

[42] G. Taguchi, S. Chowdhury, and Y. Wu. *Taguchiś Quality Engineering Handbook*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2005.

[43] J. Thackara. *How Today's Successful Companies Innovate by Design*. Gower Publishing, Hampshire, UK, 1997.

[44] D. Thomas. *Programming Ruby*. The Pragmatic Bookshelf, Dallas, TX, USA, second edition, 2006.

[45] D. Thomas and D. Heinemeier Hansson. *Agile Web Development with Rails*. The Pragmatic Bookshelf, Dallas, TX, USA, 2006.

[46] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *Computer Graphics*, 25(4):289–297, 1991.

[47] N.C. Zakas, J. McPeak, and J. Fawcett. *Professional Ajax*. Wiley Publishing Inc., Indianapolis, IN, USA, 2006.

[48] T. Zinßer, J. Schmidt, and H. Niemann. A refined icp algorithm for robust 3-d correspondence estimation. In *Proceedings of the International Conference on Image Processing*, pages 695–698, 2003.

Appendix A

Orthogonal arrays

The methodology presented in this thesis uses orthogonal arrays as a way to create samples (design concepts) based on various shape parameters. The objective here is to efficiently measure the main effects of these parameters by sampling a large input space, while still ensuring a uniform and balanced representation of it. Depending on the number of parameters and their levels, different orthogonal arrays have been presented [42] for use in Design of Experiments (DOE). Arrays with two or three parameter-levels are most commonly encountered, and in this appendix four different arrays (L_9, L_{12}, L_{18} and L_{27}) are included as examples.

The L_9-array yields nine different runs, or in this case design concepts, based on four parameters with three levels each. The L_{12}-array is specially designed to have interactions that are almost evenly distributed to all the columns. It can investigate 11 main effects with two levels each. If there are fewer parameters involved, this array can still be used by simply removing the unneeded columns. The L_{18}-array has an interaction built in between the first two columns. Interactions between three-level columns are distributed more or less uniformly among all other three-level columns, thereby permitting investigation of the main effects. This array can use a maximum of eight three-level parameters. Finally, the L_{27}-array was used for the survey of side profile shape, as described in §2.2.2. A maximum of 13 three-level parameters can be investigated by 27 unique design concepts.

Table A.1: L_9 Orthogonal array

Concept	**P1**	**P2**	**P3**	**P4**
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table A.2: L_{12} Orthogonal array

Concept	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2	2	2	2
3	1	1	2	2	2	1	1	1	2	2	2
4	1	2	1	2	2	1	2	2	1	1	2
5	1	2	2	1	2	2	1	2	1	2	1
6	1	2	2	1	2	2	1	2	1	2	1
7	1	2	2	2	1	2	2	1	2	1	1
8	2	1	2	1	2	2	2	1	1	1	2
9	2	1	1	2	2	2	1	2	2	1	1
10	2	2	2	1	1	1	1	2	2	1	2
11	2	2	1	2	1	2	1	1	1	2	2
12	2	2	1	1	2	1	2	1	2	2	1

Table A.3: L_{18} Orthogonal array

Concept	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3
4	1	2	1	1	2	2	3	3
5	1	2	2	2	3	3	1	1
6	1	2	3	3	1	1	2	2
7	1	3	1	2	1	3	2	3
8	1	3	2	3	2	1	3	1
9	1	3	3	1	3	2	1	2
10	2	1	1	3	3	2	2	1
11	2	1	2	1	1	3	3	2
12	2	1	3	2	2	1	1	3
13	2	2	1	2	3	1	3	2
14	2	2	2	3	1	2	1	3
15	2	2	3	1	2	3	2	1
16	2	3	1	3	2	3	1	2
17	2	3	2	1	3	1	2	3
18	2	3	3	2	1	2	3	1

Table A.4: L_{27} Orthogonal array

Con.	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2	2	3	3	3
5	1	2	2	2	2	2	2	3	3	3	1	1	1
6	1	2	2	2	3	3	3	1	1	1	2	2	2
7	1	3	3	3	1	1	1	3	3	3	2	2	2
8	1	3	3	3	2	2	2	1	1	1	3	3	3
9	1	3	3	3	3	3	3	2	2	2	1	1	1
10	2	1	2	3	1	2	3	1	2	3	1	2	3
11	2	1	2	3	2	3	1	2	3	1	2	3	1
12	2	1	2	3	3	1	2	3	1	2	3	1	2
13	2	2	3	1	1	2	3	2	3	1	3	1	2
14	2	2	3	1	2	3	1	3	1	2	1	2	3
15	2	2	3	1	3	1	2	1	2	3	2	3	1
16	2	3	1	2	1	2	3	3	1	2	2	3	1
17	2	3	1	2	2	3	1	1	2	3	3	1	2
18	2	3	1	2	3	1	2	2	3	1	1	2	3
19	3	1	3	2	1	3	2	1	3	2	1	3	2
20	3	1	3	2	2	1	3	2	1	3	2	1	3
21	3	1	3	2	3	2	1	3	2	1	3	2	1
22	3	2	1	3	1	3	2	2	1	3	3	2	1
23	3	2	1	3	2	1	3	3	2	1	1	3	2
24	3	2	1	3	3	2	1	1	3	2	2	1	3
25	3	3	2	1	1	3	2	3	2	1	2	1	3
26	3	3	2	1	2	1	3	1	3	2	3	2	1
27	3	3	2	1	3	2	1	2	1	3	1	3	2