Mapping Designs to User Perceptions using a Structural HMM: Application to Kansei Engineering

D. Bouchaffra, J. Tan
Department of Computer Science & Engineering
131 Dodge Hall, Oakland University
Rochester, MI, 48309, USA
Telephone: (248)370-2242
Fax: (248)370-4625
{bouchaff, jtan}@oakland.edu, http://www.cse.secs.oakland.edu/ihci/

Abstract

This paper presents a novel approach for mapping designs to user perceptions. We show how this interaction can be expressed using three classification techniques. We introduce a novel classifier called "structural hidden Markov model" (SHMM) that enables to learn and predict user perceptions. We have applied this approach to Kansei engineering in order to map car external contours (shapes) to customer perceptions. The accuracy obtained using the SHMM is 90%. This model has outperformed the neural network and the k-nearest-neighbor classifiers.

Index Terms— *User Perceptions, Engineer Designs, Structural Hidden Markov Models, Shape Modeling, Kansei Engineering.*

1 Introduction

With increasing competition and demanding consumers, engineers and designers in general are being asked to incorporate into their products[1] the elements that lead to a target set of emotions in their intended audience, the consumers. The designs with such goals are often termed *Kansei engineering* (KE) because Kansei in Japanese means a consumer's psychological feeling and image regarding a product. Kansei engineering has been popular in Japan and South Korea where it has been used in various industries such as automobiles, electrical appliances, and textile design [4, 9]. The Mazda *Miata* car is the most famous example of Kansei design. The application of Kansei engineering is composed of four main steps. The first step consists of building a subjective response vocabulary containing adjective words, i.e., the words such as "stylish", "cool", "attractive" etc., that end-users might use to describe their feelings towards the object being designed. The second step consists of building a database of subjective responses from a group of users to different products with varying characteristics. This is done by collecting users' impressions on a range of products through questionnaires using the vocabulary of step 1. The third step of Kansei engineering is to analyze the collected response data to find the latent relationships present between different feelings and design elements. In this step, linear factor analysis and multiple linear regression techniques are then used to identify those features of product designs which correlate with consumers' feelings. The final step consists of producing design rules for the development of products, and in the evaluation of prototype design solutions. It is argued, to assist in product design by ensuring that desired features are

[1]We use the terms "products" and "objects" in an interchangeable fashion to imply a *physical* object. However, the term "design" (or pattern) is used for a representation of this object.

built into products [13]. To facilitate the application of Kansei engineering, the concept of an intelligent design aid to help designers appears appealing. Such an intelligent aid could guide designers by evaluating their design in terms of the subjective responses that the design might invoke.

We propose the development of such a design aid by a methodology that is able to map subjective responses to specific components of an object or design. The methodology that we pursue in this paper uncovers latent relationships that are *more complex than linear*. Furthermore, our methodology works in an environment where the users are not forced to respond to close ended questions but are rather encouraged to provide comments freely. We believe that such an environment is more conducive to collecting "true" subjective responses. The organization of this paper is as follows: in section 2, we state the problem. In section 3, we introduce our methodology by addressing three main issues: (i) how to measure similarities between different designs and divide them into equivalent groups; (ii) how to relate subjective impressions of a design provided by a group of potential users using the electronic lexicale database WordNet [5]; and (iii) how to link the aspects of a design with different classes of subjective judgments using a novel classifier called structural hidden Markov model (SHMM). The application and experiments are discussed in section 4. Finally, the conclusion and future work are laid in section 5.

2 Problem Statement

Let $\mathcal{X}$ be the design vector space and Ω be the set of perceptions. The problem that we are trying to solve is threefold: (i) extract information that represents a design, (ii) extract the global perception $\omega \in \Omega$ assigned to a design $x \in \mathcal{X}$ based on subjective responses from users, and (iii) classify an input design into a predefined set of user perception, i.e., subjective response categories. Analytically, this third sub-problem can be expressed in the following way: determine a mapping f from $\mathcal{X}$ to Ω such that: $\forall\, x_i \in \mathcal{X}, \exists!\, \omega_j \in \Omega$ with: $f(x_i) = \omega_j$ and $dist(\omega_j, \omega^*)$ is minimum.

The class ω^* is the true perception. In other words, we attempt to predict the optimal perception ω^* assigned to the design x_i through the function f using the distance metric $dist$.

3 Proposed Methodology

Before providing the details of our proposed methodology, we first introduce some definitions.

3.1 Object Design and Perception

Three principal elements which are "design", "perception" and "classification" (or mapping) form the core of this research. Let's first define them:

Definition 3.1 *A design (representation of an object) $x \in \mathcal{X}$ is a vector defined by a pair $(\mathcal{F}, n)$ where:*

- *$\mathcal{F}$ is a set of features that characterizes the design,*
- *n is the dimension of the design feature space, ($|\mathcal{F}| = n$).*

We focus in this paper on physical designs such as an automobile. Physical designs are represented by their external shapes. Techniques of shape modeling such as Fourier descriptors, wavelet descriptors, chain codes, polygons and B-splines are being investigated [8, 2, 10, 12]. However, the challenge is to determine the one that "best" maps shapes to user perceptions.

Definition 3.2 *A perception graph $P \in \mathcal{P}$ is a triplet $(\mathcal{V}, \mathcal{U}, \mu)$ where:*

- *$\mathcal{V}$ is a collection of vertices,*
- *$\mathcal{U}$ is a collection of edges that link each pair of vertices,*
- *μ is a proximity measure that computes the strength of association between a pair of vertices.*

3.2 profile and Perception Graph

3.2.1 Distance Between Perception Profiles

Perception attributes mean different things for different individuals and can change as a function of the following variables: "the age", "the gender", "the socio-professional category" (SPC) and the cultural background (CB). Since the application domain is emphasized during the survey itself, therefore it is not a part of the profile information. These criteria constitute the *perception profile.* One way to assign values to the profile variable is for example the following:

- if age ≤ 30 then age $= 1$; else if $(31 \leq age \leq 50)$ then age =2 ; else age =3.
- if male then gender $= 0$; else gender $= 1$.
- if senior executive then SPC $= 0$; else if middle manager then SPC $= 1$; else if student then SPC $= 2$; else SPC $= 3$.
- if African then CB=1 ; else if European then CB=2 ; else if north American then CB=3 ; else if south American then CB=4 ; else CB=4.

Therefore, there is a need to consider the profile within which we conduct our survey. For example, if we collect opinions (using adjectives) of vases from young male north American students, then the profile vector is $[1, 0, 2, 3]^T$. Thus, the same adjectives used during the survey have the same meaning. Otherwise, the meanings of one word may differ subjectively if it comes from people of different age, gender, socio-professional category and cultural background. In order to measure the distance between perception attributes, one first needs to ensure that the respondents in our survey belong to the same profile. Therefore we defined a similarity measure $S(C_i, C_j)$ between two profile vectors C_i and C_j as *the number of vector components that are identical divided by the total number of variables that capture the profile.*

As an example, the similarity between the profile vectors $C_1 = [1, 0, 2, 3]^T$ and $C_2 = [2, 0, 2, 3]^T$ is equal to: $\frac{3}{4}$, a distance $D(C_1, C_2) = 1 - S(C_1, C_2)$ is induced from the similarity measure and therefore, a distance of zero means that the two perceptions are from the same profile.

3.2.2 Perception Graph and Survey

We present two different ways that enable to build the perception graph, they are:

- **Open verbal perception graph**: users are asked to give their opinions on a design such as a car or on a sub-design such as a car front. Their feelings regarding such a design are described verbally during a survey. These verbal responses could be words, phrases or sentences. In this paper, we focus only on adjectives contained within a phrase or a sentence. A speech recognizer is used in order to transcribe the uttered input into ASCII strings. A part of speech tagger is invoked in order to extract all word-forms with their respective syntactical categories. In this case, the speaker can be in a remote client site and her uttered opinion about the design is transmitted (via the Internet) to a server site where the perception graph is built (see Figure 1).
- **Closed perception graph**: in this case, predefined perceptions are used to collect responses. Each participant ranks the responses using some suitable scale. The vertices of the graph are the perception attributes that have been selected by all persons that have participated in the survey.

We have developed a similarity measure between pairs of vertices based on the electronic lexical database WordNet. WordNet accepts one word p (or perception attribute) and searches for its *synset* (set of synonyms) $W(p)$ with respect to a specified usage and profile. The following describes this similarity measure:

3.2.3 Similarity Measure between Perceptions

Here we consider how to compute a similarity between two perception attributes (words) within a specific perception profile. This measure will apply when the word is an adjective representing a vertex of the perception graph.

Definition 3.3 *The similarity measure δ of order k between two vertices p_i and p_j in a perception graph is defined as:*

$$\delta^k(p_i, p_j) = \begin{cases} \frac{|W^k(p_i) \bigwedge W^k(p_j)|}{|W^k(p_i) \vee W^k(p_j)| \times k}, if D(C_i, C_j) = 0 \\ \\ \frac{|W^k(p_i) \bigwedge W^k(p_j)|}{|W^k(p_i) \vee W^k(p_j)| \times k \times \log(1 + D(C_i, C_j))}, \quad otherwise \end{cases} \tag{1}$$

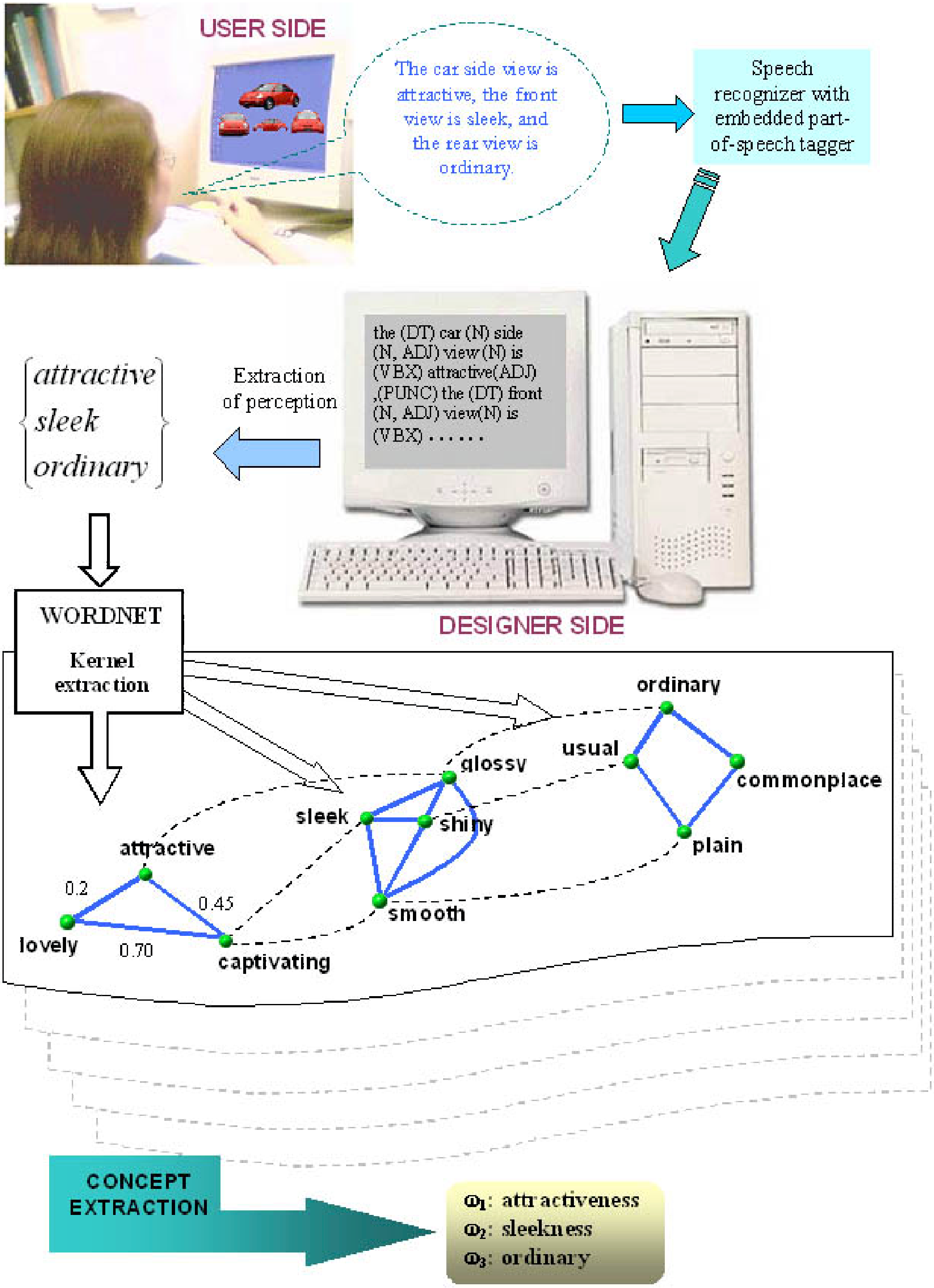

Figure 1: Three kernels and three concepts are extracted from a perception graph after a survey on "fronts" (sub-designs) of 11 different cars. The same procedure is performed for the "sides" and "rears". Through a majority vote, only one perception is assigned to a car view.

where $W^k(p) = W \circ W \circ ... \circ$ *W(p) is a set of synonyms assigned to* p *obtained by composing the WordNet operator* k *times.*

The term $\log(1 + D(C_i, C_j))$ in the denominator decreases the similarity between a pair of perceptions that are from different profiles. However, the division by k is performed in order to reduce the similarity values between perceptions that are far apart (k large!).

As an example, assume that $k = 1$, $p_1 =$"attractive", $p_2 =$ "elegant". Let's assume that the first perception is assigned to the profile $C_1^t = [1, 1, 2, 4]$ and the second perception is assigned to the profile $C_2^t = [3, 0, 0, 3]$ and let's compute δ^1(*"attractive"*, *"elegant"*). The first step is to compute the distance between these two profile vectors: $D(C_1, C_2) = 1 - S(C_1, C_2) = 1 - 0 = 1$, and $\log(2) = 0.69$. The second step consists to compute W^1(*"attractive"*) and W^1(*"elegant"*). Invoking the online lexical database WordNet, we obtain the following set of synonyms assigned to the two perception attributes:

W("attractive") = {gorgeous, captivating, handsome, beautiful, exquisite},
W("elegant") = {handsome, neat, exquisite, refined, smart.}, therefore,
W("attractive")⋀W("elegant") = {handsome, exquisite},
W("attractive")⋁W("elegant") = {gorgeous, beautiful, captivating, handsome, refined, exquisite, neat, smart.}

Therefore, the order 1 distance between these two perceptions (within particular profiles) can be written as:

$$\delta^1(\text{“attractive”, “elegant”}) = \frac{|W(\text{“}attractive\text{”}) \bigwedge W(\text{“}elegant\text{”})|}{|W(\text{“}attractive\text{”}) \bigvee W(\text{“}elegant\text{”})| \times \log(2)} = \frac{1}{4} \times \frac{1}{0.69} = 0.37$$

This value *is greater than* 0.25 which is the distance between these two perceptions but from two same profile vectors.

3.2.4 Kernels and Concepts Extraction

We introduce in this section the notions of a graph perception kernel and its concept.

- **Kernel:** is a cluster that regroups perceptions (assigned to a design or a sub-design) that are related with respect to the Wordnet-based similarity measure. The clustering process is performed using the k-means algorithm and the WordNet-based similarity measure.

- **Concept:** is the *central meaning* that is conveyed by all perceptions contained in a same kernel. In practice, the user perception (adjective) that is the closest to all other perceptions within a cluster is transformed into a noun (using Wordnet) and chosen as a concept. The set of concepts represents the set of classes that are used for a further classification task.

Figure 1 shows how we extract kernels and concepts from a verbal perception graph.

3.3 Predicting Perceptions of Physical Designs

In the first part of this section, we show how the k-nearest neighbors, and the neural networks classifiers can be used to predict the optimal concept assigned to a design (or input pattern). We also discuss their inadequacy to solve this problem. In the second part, we propose a novel classifier called *structural Hidden Markov model* (SHMM) that is more appropriate.

3.3.1 Use of Traditional Classification Techniques

- **The k-nearest neighbor approach**: we compute a distance between the design of an incoming object and the set of reference object designs stored during a training phase with their respective user perceptions. As outlined earlier, these designs are vector representation of contours of physical objects. Once a match is found (i.e., distance between two vectors is less than a threshold value ϵ!), the perception concepts $\omega_i' s$ are extracted with their respective scores. The Euclidean distance is being used in order to extract the k nearest neighbor vectors assigned to the input design.

- **The neural network approach:** the network consists of three layers: input, hidden and output, interconnected by modifiable weights. The input is a real valued vector that captures the object contour. The hidden layer has 5 units and the output has c units which are the perception concepts $\omega_1, \ldots, \omega_c$. The weights and biases are randomly initialized. We used the backpropagation algorithm which is one of the simplest and most general methods for supervised training of multi-layer neural network. This algorithm is used to determine the optimal weights for the network. The power of this algorithm allows us to calculate an effective error for each hidden unit, and derive a learning rule for the input-to-hidden weights. This rule consists of presenting a design vector and changing the network parameters to bring the actual perception concept class closer to the targeted perception concept class. Outputs are compared to the targeted values and any difference corresponds to an error. This error is some scalar function of the weights. It is minimized when the network outputs match the desired outputs. The weights are adjusted by the network to reduce the measure of the training error between ω_k^* and ω_k which are the target and the network output unit respectively. We used the cross-validation method to determine the performance of the network [3].

3.3.2 Inadequacy of Traditional Classifiers

Nonparametric classifiers are efficient when the classes (or categories) are *well defined and independent.* However, in many situations, we encounter the following:

1. The concepts (or classes) might be dependent, the dependency notion in this problem is governed by the Wordnet-based similarity measure and the concepts extraction method.

2. A class is assigned to a set of designs and not to only one. For example, there are more than one car front that are attractive.
3. An input pattern (design) might be composed of different sub-designs which are assigned to different categories. For example, an automobile front may be "attractive" whereas its rear view may be "ugly". Similarly, the visuals in a video may not be "pleasing" but the background music may be "soothing". In such cases, we would like to combine different aspects of the object in order to target a specific design.
4. A small variation of the input pattern (design) changes the class decision drastically. For example, a small deformation of a car external shape has a big impact on customer perceptions. Traditional classifiers tend to assign a same class to two "similar" input patterns.

Therefore, we introduce a novel classifier called "structural hidden Markov model" (SHMM) that takes into account the first three issues. The fourth issue is beyond the scope of this current paper.

3.3.3 The Concept of Structural HMM's

Let's consider $\omega = \omega_1, \omega_2, ..., \omega_T$ a concept sequence and $x = x_1, x_2, ..., x_T$ a design sequence. Each concept ω_i is assigned to one design x_i. For example, if the design is a car, then there are three concepts which are assigned to the car front, car side and the car rear respectively. Our problem consists of determining the optimal sequence of concepts $\omega^* = \omega_1^*, \omega_2^*, ..., \omega_T^*$ from which we can "better" observe the sequence of designs $x = x_1, x_2, ..., x_T$. Mathematically, this is written as: $\max\limits_{\omega} P(x, \omega|\lambda_{\mathcal{R}}) = \max\limits_{\omega} P(x|\omega, \lambda_{\mathcal{R}}) \times P(\omega|\lambda_{\mathcal{R}})$, where $\lambda_{\mathcal{R}}$ is the model that contains all terms involved in this computation. The equivalence relationship $\mathcal{R}$ gathers designs (or patterns) that are very similar in some sense. Therefore, $\mathcal{R}$ induces a distance (or similarity measure) in the design space. This problem is viewed as a *Structural Hidden Markov Model* (SHMM) because observations (or clusters of designs) within a hidden state (a concept) are related through a proximity measure (or an equivalence relation). In classical HMM's, the computation of the probability of an observation sequence requires the proportion of each observation within a hidden state and the state distribution [11, 1]. However, in our approach the notion of concept cannot be mapped to only one single design but to a cluster of designs that share common features. The clustering of designs is performed through the k-means algorithm on the design representation vectors. Therefore, we can write:

$$P(x_1, x_2, ..., x_T|\omega_1, \omega_2, ..., \omega_T, \lambda_{\mathcal{R}}) \equiv \; P(\dot{x_1}, \dot{x_2}, ..., \dot{x_T}|\omega_1, \omega_2, ..., \omega_T, \lambda_{\mathcal{R}}), \tag{2}$$

Let's assume as in classical HMM's that the observations are *state conditionally independent.* Hence, we finally state our problem as: determine the optimal concept sequence $\omega_1^*, \omega_2^*, ..., \omega_T^*$ such that $P(\dot{x_1}, \dot{x_2}, ..., \dot{x_T}|\lambda_{\mathcal{R}})$ is maximum. Mathematically, this can be expressed as:

$$< \omega_1^*, \omega_2^*, ..., \omega_T^* > = \arg \max_{\omega_1, ..., \omega_T} \left[\prod_{i=1}^{i=T} P(\dot{x_i}|\omega_i) \times \pi_{\omega_1} \prod_{i=2}^{i=T} P(\omega_i|\omega_{i-1}, \omega_{i-2}, ..., \omega_1) \right].$$

Therefore, we introduce the SHMM concept as follows:

Definition 3.4 *A structural Hidden Markov Chain is a quadruplet* $\lambda = (\pi, \mathcal{A}, \mathcal{B}, \mathcal{R})$, *where:*

- π *is the state initial probability vector,*
- $\mathcal{A}$ *is the state transition probability matrix,*
- $\mathcal{B}$ *is the state conditional probability matrix of the visible observations,*
- $\mathcal{R}$ *is an equivalence relation that captures the structure of the visible observations. The equivalence relation* $\mathcal{R}$ *controls the clustering process which has an impact on the matrices* $\mathcal{A}$ *and* $\mathcal{B}$ *and the initial vector* π*.*

Because of a psycho-linguistic aspect involved in the user perception process, we believe that the concept of SHMM is more adequate to solve this mapping problem. The output of the SHMM is a sequence of T concepts corresponding to each element of the whole design. A *majority vote*-based technique [7] is used in order to derive the unique optimal concept assigned to the whole design x.

3.3.4 SHMM Parameters Estimation

The structural information is embedded within the SHMM through the relation $\mathcal{R}$. As outlined above, the three parameters π, $\mathcal{A}$, and $\mathcal{B}$ are related to the clustering process created by $\mathcal{R}$. We first provide the maximum likelihood (ML) estimation of the concept conditional probability of an input design $P(\dot{x}_i|\omega_i)$. We replace each neighbor of x_i with x_i itself and each user perception (adjective) by the concept that represents it. We finally compute the number of times the design x_i was assigned to the concept ω_i divided by the number of times the concept ω_i was present in the training sample. This computation assumes that the set of designs and the set of user perceptions have already been clustered. However, the concept transition probability is ML estimated as the number of times the i-grams $< \omega_i, \omega_{i-1}, \omega_{i-2}, ..., \omega_1 >$ was present in the training sample divided by the number of times the $(i-1)$ grams $< \omega_{i-1}, \omega_{i-2}, ..., \omega_1 >$ was present in the training sample, after all the user perceptions in the training sample have been replaced by their concepts. Finally, the initial probability vector can be estimated as: the number of times the concept ω_i was assigned to an instance of x_1, divided by the total number of instances of x_1. By adding more data in the training sample, the precision of the SHMM parameters will improve. The following algorithm illustrates the parameter variation:

```
Begin
   Initialize:  k = 0; i = 0; ϵ small (close to 0) fixed value; T[i] = M (large number)
  Repeat
   ++k; ++i;
   Cluster the set of perceptions and the set of designs:  R^k
   Compute the model λ^k = (π^k, A^k, B^k, R^k)
   Compute the design maximum likelihood
        P^k = P(x_1, x_2, ..., x_T | λ^k) = P(x|ω, λ^k) × P(ω|λ^k); assign T[i] = P^k;
   Add more designs in the training sample;
 Until |T[i] − T[i − 1]| ≤ ϵ
  Store the optimal model λ^k∗
End.
```

4 Application and Experiments

We have applied this research in order to predict customer perceptions given car designs. We collected 114 images of regular cars (no trucks or vans!) with their three views (i.e., 342 images). During our survey, we have presented these images to 100 students (all young females) of Oakland University. Therefore the perception profile was specific. We used the closed perception graph method described in section 3.2. We extracted the contour x of three sides: "front (f)", "side (s)", and "rear (r)" and represent a car design as: $x = (x_f, x_s, x_r)$. In this current stage of research, we are experimenting Fourier descriptor for contour extraction. Because Fourier descriptors are not powerful enough to capture the high frequency properties (representing sharp edges), therefore other shape modeling techniques will be invoked later in this ongoing research. We have used 7 pairs of Fourier descriptors to capture the front and the rear contours and 15 pairs to describe the side contour. A feature vector is obtained through this contour extraction phase. The k-nearest neighbors, the neural networks classifiers and the SHMM have been experimented. Preliminary performance results are depicted in Table 1. SHMM outperformed the two traditional classifiers since its accuracy is 90%.

Precision (%) **Sample Size**	k-NN	NN	SHMM
70 cars	52.1	54.2	66.7
114 cars	73.2	78.6	90

Table 1: Performances obtained using the k-nearest neighbors, the neural network and the SHMM classifiers.

This optimal prediction of user perceptions within a profile is fed to the design engineer before the object (car in this application) is put into making. From the economical standpoint, our model saves a lot of money to several industrial companies.

5 Conclusion and Future Work

We have presented in this paper a novel approach that maps physical objects to user perceptions. Our methodology goes beyond Kansei-engineering since (i) the classes which are the concepts are built automatically from the user perceptions, they are not given *a-priori*, (ii) the mapping between perceptions and designs is nonlinear, thus more general. We have introduced the concept of Structural HMM as a nonlinear mapping between designs and perceptions. SHMM embeds a similarity measure (or a distance) within HMM's. However, as outlined in section 3.3.2, a small deformation of a design impacts user perceptions. As a future work, we plan to develop a more robust clustering by discovering the best partitioning of the data (in terms of preserving the pairwise similarity in the raw data). However, this investigation opens a door to category theory [6]. As for application, we plan to build an intelligent shape synthesis system that a designer is able to use and manipulate to produce automotive contours by giving commands in the form of adjectives. We believe such a system plugged into a virtual reality display will truly showcase the potential of creating designs by transforming qualitative descriptors into prototypical designs.

References

[1] D. Bouchaffra, V. Govindaraju & S. N. Srihari, Postprocessing of recognized strings using nonstationary markovian models, *IEEE Transactions: Pattern Analysis and Machine Intelligence, PAMI*, 21(10), 10 1999.

[2] C. H. Chang & C. C. Kuo, Wavelet descriptor of planar curves: Theory and applications, *IEEE Trans. Image Processing*, 5:56-70, 1996.

[3] R. Duda, P. Hart & D. Stork, *Pattern Classification*, Wiley, New York, 2001.

[4] L. Elsmark. Driving kansei in theory & practice, M.sc. thesis lith-ikp-ex-1690, Division of Industrial Ergonomics, Department of Mechanical Engineering, Linkoping Institute of Technology, 2000.

[5] Fellbaum, *WordNet: an electronic lexical database*, Bradford Book, 1998.

[6] Ulrike Hahn & Michael Ramscar, *Similarity and Categorization*, Oxford University Press, New York, 279, 2001.

[7] T. K. Ho, J. Hull & S. N. Srihari, Decision combination in multiple classifier systems, *IEEE Transactions: Pattern Analysis and Machine Intelligence*, 16(1), 1994.

[8] Jia Li & Alfred O. Hero, A spectral approach to statistical polar shape modeling, *ICIP 2002*, 9 2002.

[9] M. Nagamachi, Y. Kaneda & K. Matsushima. Automotive and kansei engineering, *Automotive Research*, 11(1):2-6, 1989.

[10] Iran Pratt, Shape representation using fourier coefficients of the sinusoidal transform, *Technical Report Series, University of Manchester*, 2-7-1996.

[11] L. Rabiner & B. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[12] T. Rachidi & L. Spacek, Boundary-based correspondence computation using the topology constraint, *5th British Machine Vision Conference*, 9 1994.

[13] L. Zhang & A. Vertiz, Kansei engineering application on commercial trunk interior design harmony, *Society of Automotive Engineers*, 2000-01-3412, 2000.