Understanding the relationships between aesthetic properties of shapes and geometric quantities of free-form curves and surfaces using Machine Learning Techniques

Aleksandar Petrov

PhD THESIS in cotutelle

To obtain the degree of

Docteur de

l'Arts et Métiers ParisTech

Spécialité "Mécanique – Conception"

École doctorale n° 432: "Science des Métiers de l'ingénieur"

and

Dottore di Ricerca della

Università degli Studi di Genova

Specialità "Ingeneria Meccanica"

Scuola di Dottorato: "Scienze e tecnologie per l'ingegneria" ciclo XXVI

Presented and defended publicly by

Aleksandar PETROV

January 25th, 2016

Understanding the relationships between aesthetic properties of shapes and geometric quantities of free-form curves and surfaces using Machine Learning Techniques

Director of thesis: **Philippe VÉRON**
Co-director of thesis: **Franca GIANNINI**
Co-supervisors of the thesis: **Jean-Philippe PERNOT, Bianca FALCIDIENO**

T
H
È
S
E

Jury
M. Jean-François OMHOVER, Associate Professor, CPI, Arts et Méties ParisTech   President
M. Umberto CUGINI, Professor, KAEMaRT, Polytechnico di Milano   Reviewer
Mme Géraldine MORIN, Associate Professor, INP, Université de Toulouse   Reviewer
Mme Rezia MOLFINO, Professor, PMAR, Università degli Studi di Genova   Examiner
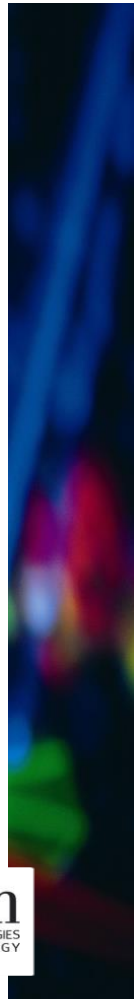
To my parents

They would have been proud

EXPLOITATION DE TECHNIQUES D'APPRENTISSAGE ARTIFICIEL POUR LA COMPREHENSION DES LIENS ENTRE LES PROPRIETES ESTHETIQUES DES FORMES ET LES GRANDEURS GEOMETRIQUES DE COURBES ET SURFACE GAUCHES
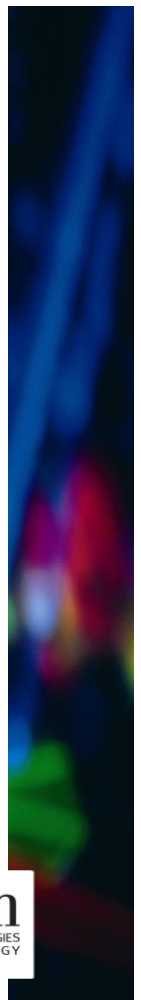
RESUME: Aujourd'hui, sur le marché, on peut trouver une vaste gamme de produits différents ou des formes variées d'un même produit et ce grand assortiment fatigue les clients. Il est clair que la décision des clients d'acheter un produit dépend de l'aspect esthétique de la forme du produit et de l'affection émotionnelle. Par conséquent, il est très important de comprendre les propriétés esthétiques et de les adopter dans la conception du produit, dès le début. L'objectif de cette thèse est de proposer un cadre générique pour la cartographie des propriétés esthétiques des formes gauches en 3D en façon d'être en mesure d'extraire des règles de classification esthétiques et des propriétés géométriques associées. L'élément clé du cadre proposé est l'application des méthodologies de l'Exploration des données (Data Mining) et des Techniques d'apprentissage automatiques (Machine Learning Techniques) dans la cartographie des propriétés esthétiques des formes. L'application du cadre est d'étudier s'il y a une opinion commune pour la planéité perçu de la part des concepteurs non-professionnels. Le but de ce cadre n'est pas seulement d'établir une structure pour repérer des propriétés esthétiques des formes gauches, mais aussi pour être utilisé comme un chemin guidé pour l'identification d'une cartographie entre les sémantiques et les formes gauches différentes. L'objectif à long terme de ce travail est de définir une méthodologie pour intégrer efficacement le concept de l'Ingénierie affective (c.à.d. Affective Engineering) dans le design industriel.

Mots clés : Courbes et surfaces gauches, techniques d'apprentissage automatiques, propriétés esthétiques, ingénierie affective, design industriel.

UNDERSTANDING THE RELATIONSHIPS BETWEEN AESTHETIC PROPERTIES OF SHAPES AND GEOMETRIC QUANTITIES OF FREE-FORM CURVES AND SURFACES USING MACHINE LEARNING TECHNIQUES

ABSTRACT: Today on the market we can find a large variety of different products and different shapes of the same product and this great choice overwhelms the customers. It is evident that the aesthetic appearance of the product shape and its emotional affection will lead the customers to the decision for buying the product. Therefore, it is very important to understand the aesthetic proper-ties and to adopt them in the early product design phases. The objective of this thesis is to propose a generic framework for mapping aesthetic properties to 3D free-form shapes, so as to be able to extract aesthetic classification rules and associated geometric properties. The key element of the proposed framework is the application of the Data Mining (DM) methodology and Machine Learning Techniques (MLTs) in the mapping of aesthetic properties to the shapes. The application of the framework is to investigate whether there is a common judgment for the flatness perceived from non-professional designers. The aim of the framework is not only to establish a structure for mapping aesthetic properties to free-form shapes, but also to be used as a guided path for identifying a mapping between different semantics and free-form shapes. The long-term objective of this work is to define a methodology to efficiently integrate the concept of Affective Engineering in the Industrial Designing.

Keywords : Free-form curves and surfaces, machine learning techniques, aesthetic properties, affective engineering, industrial design.

1   Introduction

Nowadays, it is commonly admitted that the aesthetic appearance of a product has an emphasized role in its commercial success. Today, on the market, we can find a large variety of different products and different shapes of the same product and this abundance of choices overwhelms the customer. At the beginning, the decision for buying a product is generally based on three criteria: Functionality, Price and Quality (Figure 1.1). By giving some priority to one of these criteria we can easily decide which product we are going to buy. For instance, if we take the price as a main criterion, we can find many products that satisfy the functionality with an acceptable quality for the same price. In such situation, how to make a choice? Actually, the aesthetic appearance of the product plays a key role in its commercial success. Therefore, understanding and manipulating the aesthetic properties of shapes and its visual impression in the early design phases has become a very attractive field of research.
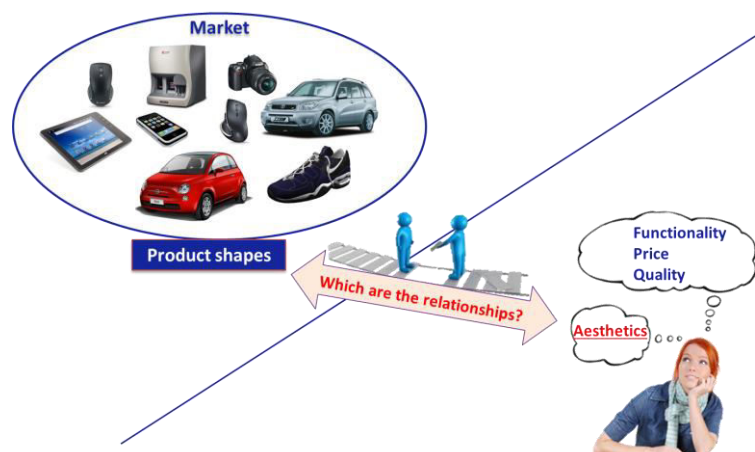

Figure 1.1: Reasoning about the relationships between the product shape and its properties

As a consequence, designing products with attractive shapes requires knowledge about the feelings and impression the products evoke on the customers that are also the

end-users. Understanding such affective influence of the product shape in the product design process requires the use of appropriate methods that can extract and transform subjective impressions about a product into concrete design parameters and tools, referred to as Affective Engineering (AE). AE is actually a new aspect integrated in the product design process that provides a platform where emotional features are incorporated into design appealing products (Nagamachi, 2011). The final and long-term objective of the AE is to define direct mapping between surface shape and emotions. Giannini et al. (Giannini & Monti, A survey of tools for understanding and exploiting the link between shape and emotion in product design, 2010) provides an overview of the most common AE methodologies applied to investigate the relationships between shape features and emotions from various disciplinary perspectives, including psychology and computer science. The objectives of this thesis are in the scope of Affective Engineering.

Affective Engineering deals with perception of shapes, which refers to very complex emotional-intuitive mechanisms that capture, organize, identify and interpret the sensory information in the nervous system. The perception is sometimes described as a process of constructing mental representations of the sensory information, shaped by knowledge, memory, expectation and attention. Regarding the classification of shapes by perception, it is common that when people classify shapes, with respect to some properties, they intuitively follow certain rules and changes of the surface shapes. Sometimes, these rules can be explicitly explained, but often they are implicit and affected by geometric properties of the shape. Aesthetic properties of shapes play a key role in the perceptual impression of shapes. However, those properties are not yet well defined for surfaces and even less mapped to surface characteristics. Moreover, trying to define the aesthetic properties and map them with surface characteristics while using classical observation techniques is practically impossible. Those mechanisms are very complex and involve many factors. Therefore, finding the direct relationships between aesthetic properties and the surface geometry requires implementation of more sophisticated methods. Having capitalized such knowledge, new industrial applications can be foreseen.

Actually, the integration of the aesthetic aspects when designing a new product has been shifted from the point of view of analysing the shapes with respect to production aspects towards the user's acceptance point of view. In the past, when the aesthetic appearance of the product was less relevant, all products were designed taking into account production and cost constraints. Therefore, designers could not pay enough attention to the shape they prefer, because they had to focus on the shape that could be reached considering the available manufacturing equipment. It was a closed system (designers-manufacturers) where the customers were involved only at the beginning and at the end. Their interests and tastes were captured at the beginning, and at the end, they were verifying the product compliance. Additionally, in the current competitive market, being faster in reaching the customers' expectations is becoming extremely important. Therefore, we can foresee a new product development process in which the customer is an important actor

alongside with the other experts in analysing and defining the product. Taking into consideration the fact that customers are valued more than before, the aesthetic aspects of the product are becoming significantly more important and a key factor for the acceptance of certain products on the market. Now, with the availability of new materials and the development of new manufacturing technologies such as 3D printing (Additive Manufacturing facilities) and five-axis CNC machines, we do not pose the question of which shape can be produced, but which shape we want to be produced anymore. Thus, not only do designers have more 'freedom' to design what they like, but also users can play the role of designers and produce products they designed themselves. Low-cost 3D printers provide interesting infrastructures to manufacture products designed by non-professional designers. In order to help non-professional designers to design their products, CAD systems need to be more intuitive and to have user-oriented design tools and parameters integrating an interaction language closer to non-professional designers.

For example, the qualitative judgment of a shape from a non-professional point of view often considers more abstract and general notions (e.g. words) to describe the shape. These words, further, can be used to define high-level manipulation tools. The development of geometric modeling systems allowing the users to employ previously defined words to construct the desired shape is called Declarative Modeling (Lucas, Martin, Philippe, & Plémenos, 1990). Its main advantage is the ability to allow the creation of objects by providing only a set of abstract words, generally based on geometric, topological or physical properties, widely known among non-professional designers. Such creation of objects requires identification of the relationships between the abstract word meaning and geometric characteristic of the shape. This is even much more difficult when the aim is to map emotions to geometric model. The emotional description of the shape is very difficult and ambiguous task because it depends on personal knowledge, experience and culture. Therefore, the mapping between emotions and the surface shapes is improved by inserting an intermediary layer (aesthetic layer). It is more feasible, firstly, to discover which aesthetic properties evoke certain emotions, and then finding the relationships between the aesthetic properties and the geometric shape in order to map the emotions to the surface shape.

Considering the customer's needs in the design process emphasizes the difference of the languages used in different activities of the design process (Giannini, Monti, & Podehl, Aesthetic-driven tools for industrial design, 2006). The description of the aesthetic characteristics of a shape is made by using an appropriate vocabulary, which can be also considered as a separate language. The following figure (Figure 1.2) shows the layout of the main levels of shape description that also represents the three languages used in the different phases of the Product Design Process (PDP).

The first level of the shape description is the geometric level where the surface shape is represented by means of mathematical models, e.g. NURBS curves and surfaces. At the geometric level, shapes are characterized by classical geometric quantities of the surface, e.g. sur-

face area or curvature evolution. The second level of shape description is the aesthetic level where shapes are characterized by aesthetic properties, e.g. flatness or convexity. The final level is the emotional level where the surface shape is described in terms of what emotions it evokes to the customer. This level of shape description corresponds to the so-called marketing language described in (Giannini, Monti, & Podehl, Aesthetic-driven tools for industrial design, 2006).
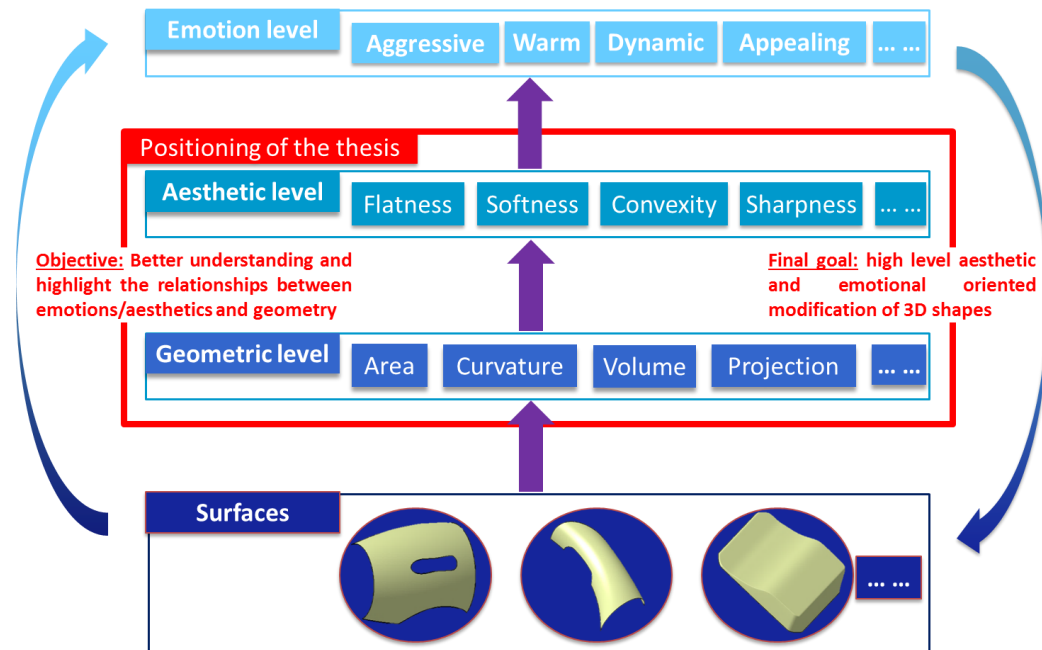

Figure 1.2: Mapping between a surface shape with emotions through aesthetics

The aesthetic character of the shapes affects the evoked emotions. However, stylists and designers do not explicitly encode them in their models. Therefore, a more reliable mapping between the product shape and the emotions is the mapping that is carried out while considering the aesthetic properties of the shape. This is the approach followed in the European Project FIORES – II (Character Preservation and Modelling in Aesthetic and Engineering Design (FIORES-II, 2000).

This thesis introduces **a generic framework for mapping aesthetic properties to 3D free-form shapes by using machine learning techniques to extract aesthetic classification rules and associated geometric parameters**. Thus, only the two first levels of Figure 1.2 are considered. By extracting aesthetic classification rules from a dataset provided through interviews of non-designers (potential customers), we understand which surface characteristics are directly related to the aesthetic properties. Thus, the discovered knowledge opens new perspectives for designing **high-level tools to manipulate** geometric models. These tools correspond to the modification of shapes described by high-level shape descriptors using few parameters that are close to the way designers, stylists but also customers think.

The contribution of this thesis can be better interpreted as shown in Figure 1.3, where Machine Learning Techniques are used to fill (bridge) the gap between geometric models and aesthetic properties. The proposed framework provides tools for mapping aesthetic properties to 3D free-form shapes by extracting the classification rules and associated shape parameters. The main idea relies on the application of learning algorithms that are able to discover hidden knowledge and classification patterns in a dataset. Of course, in order to be able to do this, a set of many other activities have been carried out to obtain the dataset. Some of the more important activities are: defining the target shapes and objects, creation of the initial dataset (IDS), development of an environment (GUI) for carrying out interview, capturing the classification of the participants, defining surface parameters for detecting the rules, selection of the learning algorithms, and application of the best performant learning algorithm. The necessity for applying the learning algorithms lies in the fact that it is almost impossible to find a direct correlation between the aesthetic properties and the geometric quantities without using Artificial Intelligence techniques, such as Machine Learning Techniques (MLTs). There is large number of geometric quantities that can be computed for a surface, so it is evident that mapping between the aesthetic properties and surface shape is not possible without using computer algorithms. The selection of the most relevant shape parameters, with respect to an aesthetic property, can be further used as aesthetic parameters, enabling aesthetic-oriented manipulations. Those high-level manipulations are not directly addressed in this document, i.e. we do not try to modify shapes using aesthetic parameters, but the knowledge that is capitalized here can be used to define such higher-level operators.

The most important thing is that the framework is generic and can be applied to identify a mapping between different semantics and free-form shapes. Thus, it is suitable for application in various domains and for analysing other aesthetic properties. It can be considered as a guided path for structuring and understanding aesthetic properties of shapes.
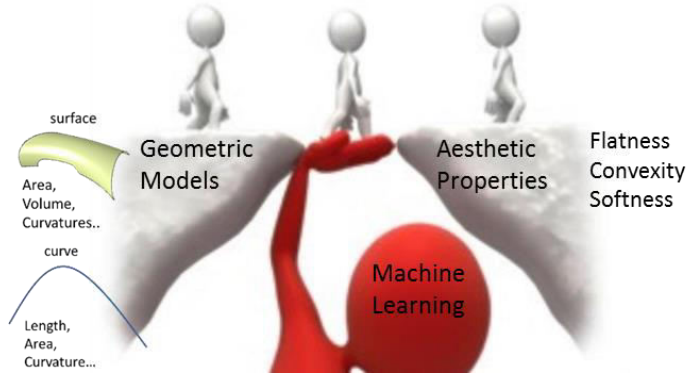

Figure 1.3: Our goal: bridging the gap between geometric models and aesthetic properties using MLTs

This thesis is organised into two main parts: state-of-the-art (part A) and proposed framework (part B). The study of the state-of-the-art introduces the different research areas that are basic elements of the generic framework. It has been decomposed in three chapters.

Chapter 2: It introduces the most common methods of geometric representation for geometric modeling in product design. The parametric representation of geometric entities has proven to be the most suitable in geometric modeling. Furthermore, the properties of the most widely used parametric representations, the Bézier, B-spline and NURBS curves and surfaces, have been elaborated. This chapter also introduces the different types of geometric modeling strategy that are used in the design process activity. By moving from the question on how to design the shape to the question what shape we want to design, we move from procedural towards declarative designing techniques. This chapter ends with underlying the needs for a more intuitive modification of the geometric models.

Chapter 3: This chapter introduces the necessity of having high-level modification tools. Declarative modeling introduces the concept of using more abstract notions based on geometric, topological, physical or aesthetic properties to model objects. There are two general approaches to associating aesthetic properties to free-form shapes: mapping of aesthetic properties and definition of Aesthetic Curves and Surfaces. The latter approach is curvature based, less intuitive, and inconvenient in application for designing more complex shapes. Therefore, the approach of mapping of aesthetic properties have been adopted and the results of the FIORES II project have been considered as a good starting point for mapping aesthetic properties to surfaces.

Chapter 4: This chapter presents the Data Mining methodology and the most widely used Machine Learning Techniques. Depending on the numbers of labels, the supervised learning problem is divided in two major groups, the single-label (one label) and multiple label (more than one label) learning problems. The single-label learning is considered the standard task, but not less natural and less intuitive is the multi labelled learning. The human brain can naturally associate one idea with multiple classification concepts. Since the perception of aesthetic properties represents a type of brain activity, the single-label and multiple labelled classification problem solving algorithms are summarized in this chapter. The actual mapping of aesthetics to any geometric entities consists of finding which surface geometric quantities are highly correlated with respect to a given aesthetic property. Unlike curves, the surfaces are much more complex geometric entities and the number of geometric quantities that can be computed is very large. It seems to be almost impossible to find mapping between surface quantities and aesthetic properties without using Artificial Intelligence.

Part B aims at presenting the contribution of the thesis by introducing the proposed generic framework that maps aesthetic properties (Chapter 3) to 3D free-form shapes (Chapter 2) using Machine Learning Techniques (Chapter 4) in order to extract aesthetic classification rules and associated geometric quantities. Part B is also organized in three chapters:

Chapter 5: This chapter introduces the overall generic framework and its constituent parts. The chapter also describes framework validation carried out on 2D curves. Such a validation has been possible due to the existence of a complete definition of the measure of straightness for curves and their classification in different classes according to different ranges of the straightness measure. The results obtained from the validation step gives the right to state that MLTs are good at identifying classification rules and associated relevant geometric quantities. These results validate the proposed framework that can then be applied in mapping aesthetic properties to surfaces.

Chapter 6: This chapter introduces the challenges of applying the framework for identifying the mapping of aesthetic properties to the surfaces. Since surfaces are much more complex geometric quantities than curves, the setup of the framework requires additional steps of data acquisition, pre-processing and preparation. The acquisition of the data has been carried out by interviewing group of non-designers people. In order to expedite the classification process and to reduce the classification time, a GUI in Matlab has been created. During the interview, the participants were requested to classify sets of surfaces in four classes (Flat, Almost Flat, Not Flat, and Very Not Flat). This chapter answers to the main questions of the thesis concerning whether there is a common judgment for the flatness; how the surrounding affects the perception of flatness; how the position and the transition of a surface toward the surrounding affect the perception of flatness; and which surface parameters are relevant with respect to the flatness.

Chapter 7: This chapter gives the overall conclusion and underlines the scientific contribution of the thesis. The discussion part of the results is included along with new perspectives for the application of the framework in the future.

Chapter 2

Geometric modeling in product design

This chapter presents an overview on the basic geometric modeling methods used in the design processes. The most common geometric representation methods are introduced at the beginning of this chapter (Section 2). Furthermore, after comparing the most widely used representation methods, a more detailed presentation of the parametric representation method is given (Section 2.2). This section is divided in two subsections, presenting the Bézier, B-Spline and NURBS curve models (2.2.1) and the corresponding surface models (2.2.2). The third section (2.3) presents the different types of geometric modeling strategies that are part of all CAD systems. This section introduces the surface modeling in product design (2.3.1) and surface models which are considered a medium for transferring the aesthetic properties to the customers. Further down in the same section, the basics of the procedural design process (2.3.2) are presented. This section ends by presenting the needs for intuitive modification of the geometric models (2.3.3). The last section (2.4) gives a synthesis of this chapter.

2. Geometric modeling in product design activities

2.1 Geometric representation methods

The most common methods of representation of geometric entities (curves and surfaces) in geometric modeling are *parametric*, *implicit* and *explicit* methods. The explicit methods (Foley, van Dam, Feiner, & Hughes, 1992) that express, for example, y = f(x) (and z = g(x,y) for surfaces) are quite limiting. For instance, it is impossible to get multiple values of y for a given x, hence circles and ellipses must be represented with multiple curve parts. Furthermore, such representations are not rotationally invariant and describing curves with vertical tangent is difficult, because a slope of infinity is difficult to be represented. This leaves us with parametric and implicit methods as the two key approaches for geometric modeling tasks. An implicit representation (Hoffmann, 1993) of a surface is in the form $F(\boldsymbol{p}) = f(x, y, z) = 0$ (for instance, $x^2 + y^2 - r^2 = 0$ for a circle or $x^2 + y^2 + z^2 - r^2 = 0$ for a sphere), where $\boldsymbol{p}$ is a point on the surface implicitly defined by the expression $F(\boldsymbol{p})$. If the relation $F(\boldsymbol{p}) = 0$ holds for $\boldsymbol{p}$, it means that the point $\boldsymbol{p}$ is on the surface. This representation provides the possibility for testing whether a given point $\boldsymbol{p}$ is on the surface or not. However, this representation does not provide a direct way for systematically generating points on the surfaces.

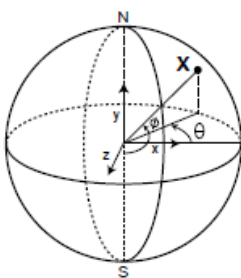Usually, an implicit representation is constructed so that it has the property that if for given *x*:

$F(p) > 0$, then the *x* is "above" the curve (surface)

$F(p) = 0$, then the *x* is exactly on the curve (surface)

$F(p) < 0$, then the *x* is "below" the curve (surface)

Thus, an implicit representation allows for a quick and easy inside-outside (or above – below) test. Additionally, the implicit methods provide mathematical tractability very suitable in computer animation and for modeling operations such as blending, sweeping, intersection, Boolean operations. The implicit methods can be used for modeling any arbitrary closed surfaces as a single patch, but for such created surfaces are less interactive because of the possible self-interactions and other singularities (Bajaj, Chen, & Xu, 1994; Bajaj, Chen, & Xu, 1994). Such limitations make implicit surfaces not suitable for product design, where the fine control of the shape and its conversion to other representation for simulation purposes are necessary. This motivates the large adoption of parametric methods (Hoffmann, Geometric and Solid Modeling: An Introduction, 1989), in commercial CAD (Computer-Aided Design) systems. Other properties that motivate the parametric methods of representation to be widely used in CAD systems are their coordinate system independence, vector-valued function, ease of handling vertical slopes, and (the most important) efficient generation of points on a curves (surfaces) which is crucial for shape representation in product design. In the parametric method, each of the coordinates of a point on the curve (surface) is represented separately as an explicit function of an (two) independent parameter(s).Thus, parametric curves and surfaces are defined as follows:

P(u) = (x(u), y(u)) for curves or P(u,v) = (x(u,v), y(u,v) z(u,v))   for surfaces with a ≤ u,v ≤ b

For example for a sphere u = φ and v = ϑ

$x(\varphi, \vartheta) = r\, sin(\varphi)\, cos(\vartheta),$

$y(\varphi, \vartheta) = -r\, cos(\varphi),$

$z(\varphi, \vartheta) = -r\, sin(\varphi)\, sin(\vartheta), \qquad 0 \le \vartheta \le 2\pi\ and\ 0 \le \varphi \le \pi$

Where, *P(u)* (or *P(u,v)*) is a vector-valued function of the independent variable *u* (or u,v). Although the interval [a, b] is arbitrary, it is usually normalized to [0, 1]. Unlike the implicit representation, the parametric representation allows us to directly generate points on the curve (surfaces). All that is required is to choose the values of the parameters *u* and *v* and then *P(u, v) = F(u, v)*. If it is done in a systematic way over the range of possible *u* and *v* values, it is possible to generate a set of points sampling the entire curve (surface). The main

shortcoming of the parametric curve (surface) is the inability of positioning an arbitrary point in space in terms of testing whether the point is on the curve (surface).

Piegl and Tiller, in the NURBS book, summarize and list the advantages and disadvantages of the parametric representation method (Piegl & Tiller, The NURBS Book, 1997) and the most significant are listed in the following:

1. The parametric method is easy to extend from two-dimensional space into three-dimensional space by simply adding a z coordinate.
2. The parametric method is very practical to represent bounding curve segment (or surface patches)
3. Parametric curves (surfaces) have ordering of the sequence that is gained from the ordering of the parameter '*u*' in the knot vector (u and v for surfaces). Such sequence ordering helps generating ordered sequences of points of a curve.
4. From computational point of view, the parametric method is much more convenient due to the numerically stable algorithms used to represent the geometry in a computer
5. Computing a point on a curve or surfaces is an easy task in the parametric method but testing whether the point is on a curve or surface is very difficult in the parametric method
6. The parametric method sometimes has to deal with parametric anomalies which are related more to the computation algorithm imperfection. For instance, the poles of a sphere are same as the other points on the sphere, but are algorithmically difficult.

Since both parametric and implicit forms are complement to each other in different application, it is very practical to transform one form to the other. The possibility of having mutual conversion helps combining their advantages in order to obtain better geometric representation of shapes. The conversion from parametric to implicit form is known as implicitization, whereas the inverse conversion is known as parametrization (Sederberg, Anderson, & Goldman, 1984).

2.2   Parametric representations

The parametric representation of the geometric entities has been adopted by almost all CAD systems enabling all analytic shapes to be represented, as well as more complex free-form entities. The Bézier, B-Splines and NURBS representations are the most used in free-form curve and surface definition (Piegl & Tiller, The NURBS Book, 1997), and they are presented in the following sections.

2.2.1 Bézier, B-Spline and NURBS curves

2.2.1.1 B-Spline curves

The basics of the Bézier curves have been introduced in the theoretical mathematics long time before their implementation in computer graphics by the French mathematician Charle Hermite and the Russian mathematician Sergei Bernstein. Later, the mathematical formulation of the Bézier curves has been proposed by the French engineer Pierre Bézier at Renault Car Company.

There are many ways of interpreting the Bézier curves depending on which aspect is considered, the engineers prefer to interpret the Bézier curves in terms of the center of mass of a set of point masses. For instance, we can consider four masses (m_0, m_1, m_2, and m_3) positioned at four points P_0, P_1, P_2, P_3 (Figure 2.1). The position (P) of the center of mass is given by the following equation:
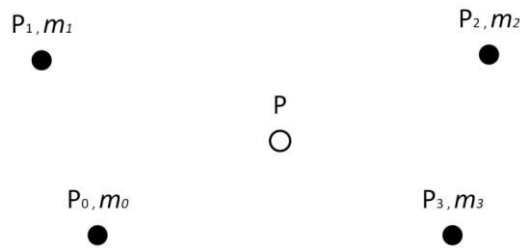
Figure 2.1: Center of mass of four points

$$\mathbf{P} = \frac{m_0 P_0 + m_1 P_1 + m_2 P_2 + m_3 P_3}{m_0 + m_1 + m_2 + m_3} \qquad (2.1)$$

The position (P) of the center of mass remains unchanged if the masses at each point are constant. Supposing that instead of keeping the masses constant, each mass varies according to some function of a given parameter 'u'. For example, assuming that the masses at each points vary according following functions: $m_0 = (1-u)^3$, $m_1 = 3u(1-u)^2$, $m_2 = 3u^2(1-u)$, $m_3 = u^3$. The values of these masses as a function of 'u' are shown in this graph (Figure 2.2):
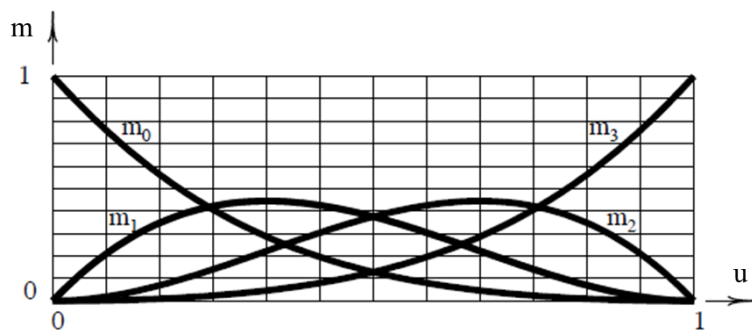
Figure 2.2: Cubic Bézier blending functions

If the value 'u' varies between 0 and 1 then the position of the center of mass (P) changes continuously sweeping out a curve. This curve is called a cubic Bézier curve (Figure 2.3). These masses (m_i) function are called *blending functions* whereas their positions (P_i) are known as *control points* of the Bézier curves. The lines connecting two neighboring control points create a figure know as *control polygon.* In the case for Bézier curves, the blending functions are also known as *Bernstein polynomials.*
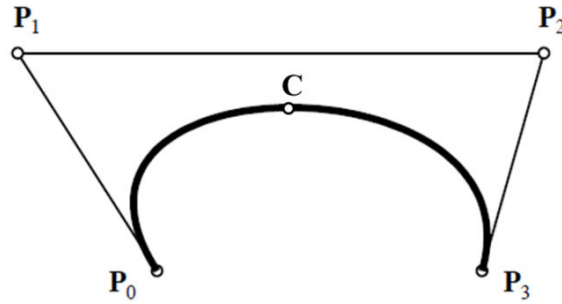
Figure 2.3: Cubic Bézier curve

Bézier curves of any degree can be defined; Figure 2.4 shows set of Bézier curves of different degree
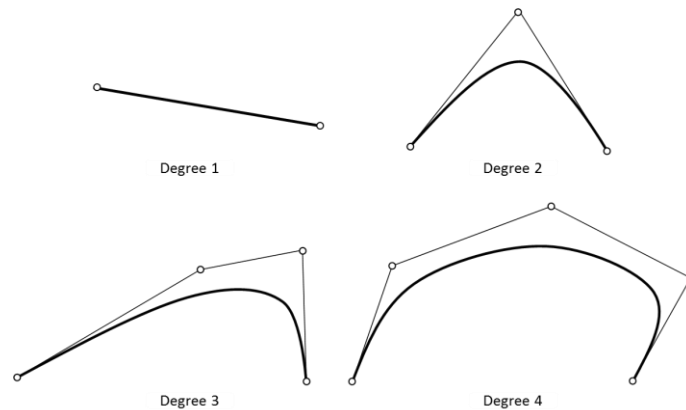
Figure 2.4: Bézier curves of various degrees

A degree n Bézier curve has n + 1 control points whose blending function are denoted $B_i^n(u)$ is referred to as ith Bernstein polynomial of n degree, where

$$B_i^n(u) = \binom{n}{i}(1-u)^{n-i}u^i, \ i = 0, 1, 2, \dots, n. \tag{2.2}$$

$\binom{n}{i}$ is called a *binomial coefficient*, equal to $\frac{n!}{i!(n-i)!}$. Thus, the equation of a Bézier curve is:

$$C(u) = \sum_{i=0}^{n}\binom{n}{i}(1-u)^{n-i}n^iP_i \tag{2.3}$$

$$C(u) = \sum_{i=0}^{n}B_i^n(u)P_i \tag{2.4}$$

Bézier curves are widely used in computer graphics to design free-form curves. The usefulness of Bézier curves is due to their many geometric and analytical properties. These properties are either implied directly from de Casteljau's algorithms or properties of the Bernstein polynomial. Some of the most important Bézier curves properties are listed below (Farin, 1996):

1. **Affine invariance.** Bézier curves are invariant under the common CAD transformation such as rotation, translation and scaling.
2. **Invariance under affine parameter transformation.** Usually, the parameter *'u'* varies between 0 and 1, hence, the transformation from this interval to any arbitrary interval *a* and *b* is called invariance under affine parameter transformation.
3. **Convex Hull.** This property refers more to the position of a Bézier curve with respect to the control polygon and states that it will always be completely laying inside of the convex hull of the control polygon.
4. **Endpoints interpolation.** The Bézier curve always begins from the first control point and ends at the last control point.
5. **Symmetry.** Inverting the order of the control points will not affect the shape of the curve and will produce the same curve.
6. **Pseudo-local Control**. This property refers to the inability of having local changing of the shape and the entire curve is affected by moving one control point.

The Bézier representation has two main disadvantages. First, the degree (*n*-1) of the Bézier curve directly depends on the number of control points (*n*), so if we want to model a complex shape using the Bézier curve, then its Bézier representation will have a prohibitively high degree (for practical purposes, degree exceeding 10 are prohibitive). The high degree of curves is inefficient to process and is numerically unstable. Moreover, from manipulation point of view, modeling a complex shape using a single-segment Bézier curves is not well-suited to interactive shape modification. Although Bézier curves can be shaped by means of their control points, the control is not sufficiently local and changing one control point affects the entire curve, making the design of a complex shape a very difficult task. Such complex shapes, however, can be modeled using composite Bézier curves named *B-spline* curves to obtain piecewise polynomial curves.

2.2.1.2   B-Spline curves

The limitations of the Bézier curves are overcome by formulating of the B-spline curves which are considered as a generalization of the Bézier curves. In fact, B-spline curves have more desired properties than Bézier curves gained by undertaking the Bézier curves properties and overcoming the two main limitations of the Bézier curves. Namely, replacing a Bézier curve of higher degree with a series of many Bézier curves (B-spline) of lower degree

reduces the overall degree of the curve and improves the property of local modification. A B-spline curve of degree p with n control points consists of a series of $n - p$ Bézier curve segments. For instance, a cubic curve (degree p = 3) with n = 7 control points has 4 Bézier curve segments (Figure 2.5) and these segments all have C^2 continuity at the join points. The continuity degree is in terms of curve degree and repetition of knots in the knot vector. In general, it can be stated that the B-spline curve requires more computation, but is far more flexible and pleasant to work with, which is the reason why it has become part of almost every serious graphics development environment.
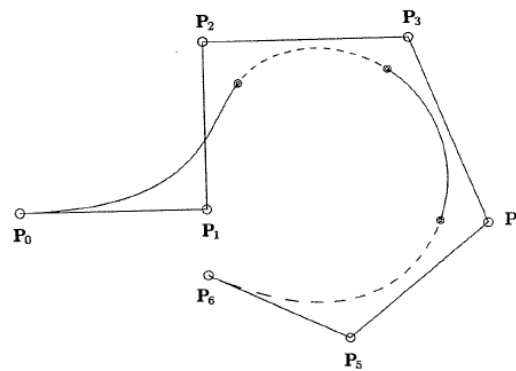
Figure 2.5: **Cubic B-spline curve composed of four Bézier curve segments (Piegl & Tiller, The NURBS Book, 1997)**

Before presenting the B-spline curve, first of all, it is necessary to define both the knot vector and the B-spline basis function. The joint points between these segments are called *knots* and they play a fundamental role in the understanding of this kind of curves. Considering $U = [u_0, ..., u_m]$ to be a non-decreasing sequence of real numbers, that is $u_i \leq u_{i+1}$, $i = 0, ..., m$; where u_i are called *knots*, and U is referred to as the *knot vector*. The knot vector is used to specify values in the evaluation interval where the curve changes the spline segment. By spacing the m intervals of the knot vector with equal distance, we obtain a *uniform B-spline curve* (for instance, $U = \{0,0,0,0,2,4,6,8,8,8,8\}$), while an uneven spacing (for instance, $U = \{0,0,0,0,1,5,6,8,8,8,8\}$) yields a *non-uniform B-spline curve.* Repeating a knot in the knots vector (i.e. increasing the multiplicity) reduces the continuity of the curve at that knot. Thus, repeating the knots p times at the ends of the knots vector will make the B-spline to pass through the endpoints of the control polygon (for instance, $U = \{0,0,0,0,2,4,6,8,8,8,8\}$, the 0 is repeated three times same as the 8) .

When constructing a uniform B-spine curve, the basic functions translate each other, yielding a very simple basis function. The Cox de Boor algorithm can be found below, which can recursively computes any uniform (Figure 2.6) or non-uniform (Figure 2.7) B-spline basis function of n degree.

$$N_{i,0}(u) = \begin{Bmatrix} 1 & if\ u_i\ \leq u\ <\ u_{i+1} \\ 0 & otherwise \end{Bmatrix} \tag{2.5}$$

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i}\,N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}\,N_{i+1,p-1}(u) \tag{2.6}$$
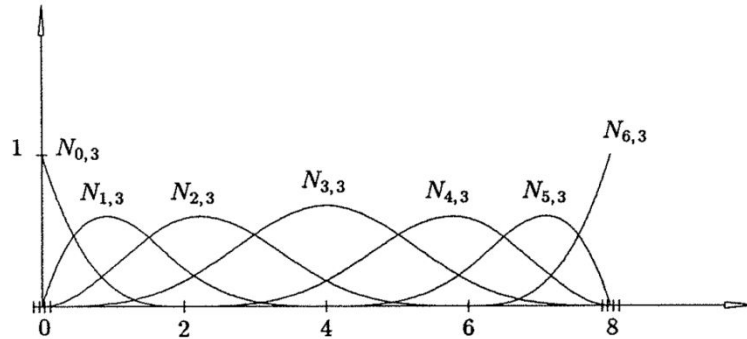
Figure 2.6: Uniform cubic basis function defined on U = {0,0,0,0,2,4,6,8,8,8,8} (Piegl & Tiller, The NURBS Book, 1997)
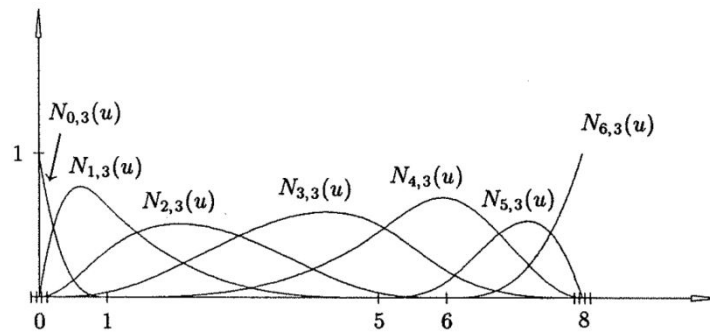
Figure 2.7: Non-uniform cubic basis function defined on U = {0,0,0,0,1,5,6,8,8,8,8} (Piegl & Tiller, The NURBS Book, 1997)

As mentioned before, the B-spline curves share their properties with the Bézier curves, which are listed in the corresponding section. For more detailed information about the B-splines properties, the readers are invited to refer to The NURBS book (Piegl & Tiller, The NURBS Book, 1997). Regarding the Bézier curve, a B-spline curve requires some more additional information (i.e. a knot vector and the degree of the curve) and its mathematical definition is a bit more complex than Bézier curves. But, it has more advantages to compensate this shortcoming. Firstly, the degree of the B-spline is independent from the number of control points which was the biggest shortcoming of the Bézier curves. The use of the knots vector results in providing greater local control flexibility. In other words, by overcoming the limitations of the Bézier curve, we can create a B-spline curve of lower degree using large number of control points, thus supporting local modification. However, considering the fact that B-spline curves are still polynomial, they cannot be used to represent analytical curves (e.g. circles and ellipses). Therefore, a generalization of B-spline is required and introduced in the following section.

2.2.1.3 NURBS curves

The mathematical definition of NURBS curves is relatively simple. A NURBS curve is a vector valued piecewise rational polynomial function of the form:

$$C(u) = \frac{\sum_{i=0}^{n} w_i N_{i,p}(u) P_i}{\sum_{i=0}^{n} w_i N_{i,p}(u)} \qquad a \leq u \leq b \qquad (2.7)$$

where $N_{i,p}(u)$ - normalized B-spline basis function of degree p, the P_i - control points, the w_i are the weights, and u_i are the knots of a knot vector. The knot vector, for non-uniform and non-periodic B-spline, has the following form:

$$U = \{a, ..., a, u_{p+1}, ..., u_{m-p-1}, b, ..., b\}$$

where the knot vector end's with a and b which are repeated with multiplicity of $p + 1$ and in most practical application $a = 0$ and $b = 1$. The degree, number of knots, and number of control points are related by the formula $m = n + p + 1$. The NURBS curves share all important properties of the B-splines and Béziers curves, but for more detailed information concerning the property of NURBS, the reader can refer to (Piegl, On NURBS: A survay, 1991)

2.2.2 Bézier, B-Spline and NURBS surfaces

2.2.2.1 Bézier surfaces

The Bézier curve **C(u)** is a vector-valued function of one parameter *'u'*. A Bézier surface is a vector-valued function of two parameters, *u* and *v*, and represents a mapping of a region, in a (*u,v*) plane parametric space into Euclidean three-dimensional space. **S**(*u,v*) = (*x(u,v)*, *y(u,v)*, *z(u,v)*), *u,v* ∈ **R**. The *tensor product scheme* is probably the simplest method, and the one most widely used in geometric modeling application.

The tensor product method is basically a bidirectional curve scheme. It uses basis function and geometric coefficients. The basic functions are bivariate function of *u* and *v*, which are constructed as product of univariate basis functions. The Bézier surfaces (Figure 2.8) are obtained by taking a bidirectional net of control points and products of the univariate Bernstein polynomials:

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u)\, B_j^m(v)\, P_{i,j} \quad 0 \leq u,v \leq 1 \tag{2.8}$$
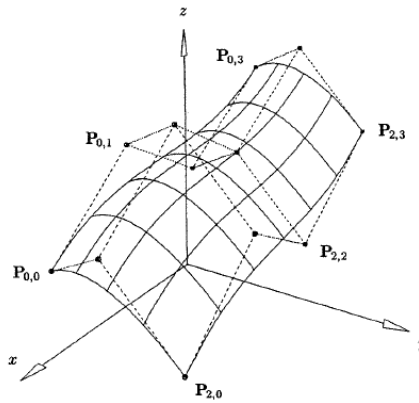
Figure 2.8: A quadratic x cubic Bézier surface

The Bézier surfaces share all important properties and limitations with the Bézier curves.

2.2.2.2 B-spline surfaces

A B-spline surface is obtained by using a bidirectional set of control points, two knot vectors (U and V), and the product of the univariate B-spline function. The B-spline tensor product surface is defined with the same parameters as B-spline curve, but every parameter is doubled, for both the *u* and *v* direction Figure 2.9.

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\, P_{i,j} \quad 0 \leq u,v \leq 1 \tag{2.9}$$
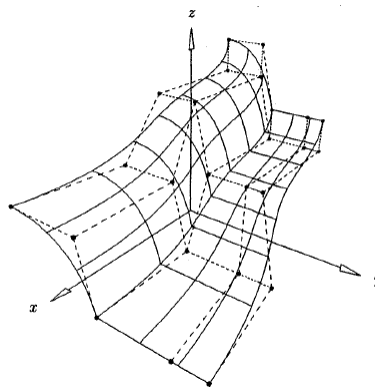
Figure 2.9: A quadratic x cubic B-spline surface with U = {0,0,0,1/2,1/2,1,1,1} and V = {0,0,0,0,1/2,1,1,1,1}

Properties and limitations of the B-spline surfaces are similar to properties of B-spline curves, for each (*u* and *v*) direction separately (for more details see (Piegl & Tiller, The

23

NURBS Book, 1997). B-splines are a particular case of NURBS where all weights are equal to 1. The B-splines are far more stable and faster in representation of intersection but they are not able to represent analytics exactly. So, this is the reason why the mathematical formulation and properties of the NURBS have been presented below.

2.2.2.3 NURBS surfaces

A NURBS surface is the rational generalization of the tensor product non-rational B-spline surface, but itself, the NURBS surface is, in general, not a tensor-product surface. A NURBS surface is a bivariate vector valued piecewise ration function of the form:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\, w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\, w_{i,j}} \boldsymbol{P}_{i,j} \quad 0 \le u, v \le 1 \tag{2.10}$$

where p and q are the degree in the u and v direction respectively, $\boldsymbol{P}_{i,j}$ form a bidirectional control net, $w_{i,j}$ are the weights, and the $N_{i,p}(u)$ and $N_{j,q}(v)$ are the non-rational B-spline basis function defined over the knot vectors:

$$U = \{0, ..., 0, u_{p+1}, ..., u_{r-p-1}, 1, ..., 1\}$$

$$V = \{0, ..., 0, v_{q+1}, ..., v_{s-q-1}, 1, ..., 1\}$$

where the end knots are repeated with multiplicities of $p + 1$ and $q + 1$, respectively, and the degrees, number of knots, and number of control points are related by the formulas $r = n+p+1$ and $s = m+q+1$. NURBS surface can be analyzed similarly using the bivariate rational basis function:

$$R_{i,j}(u,v) = \frac{N_{i,p}(u)\, N_{j,q}(v)\, w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\, w_{i,j}} \tag{2.11}$$

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)\, \boldsymbol{P}_{i,j} \tag{2.12}$$
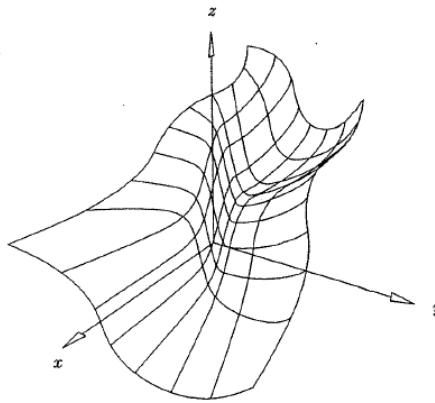
Figure 2.10: Bicubic NURBS surface defined by the U = V = {0,0,0,0,1/2,1,1,1,1}, $w_{1,1}$ = $w_{1,2}$ = $w_{2,1}$ = $w_{2,2}$ = 10, $w_{i,j}$ = 1 with I,j ≠ 1,2

In fact, the important properties of the $R_{i,j}$ (*u,v*) are inherited from those of the non-rational basis function $N_{i,p}$ (*u*) and $N_{j,q}$ (*v*) and the overall properties of the NURBS surface are the same as the properties of NURBS curves.

NURBS (curves and surfaces) are standard methods for the representation and design of shapes in all CAD systems. Some reasons for the widespread acceptance and popularity of NURBS in the CAD systems and graphics community are as follows:

- The mathematical formulation of the NURBS provides the possibility for representing both the analytic shapes (conics. quadrics. surfaces of revolution. etc.) and free-form shapes
- NURBS provide the possibility to modify the shape by manipulating both the control points and the weights.
- From the numerical computation point of view, their computation is reasonably fast and stable.
- NURBS have understandable geometric interpretations, making them useful for those designers who have a very good knowledge in technical drawings.
- Powerful capabilities have been defined for NURBS (such as knot insertion/refinement/removal, degree elevation, splitting, etc.), which can be used throughout to design, analyze, process, and interrogate objects.
- NURBS, same as the other representation tools, are affine invariant under scaling, rotation, and translation.

The drawbacks of the NURBS are following:

- The definition of a NURBS requires much more storage than it is needed to define traditional curves and surfaces. For instance, the representation of a circle using NURBS requires 10 knots and 7 control points, whereas traditional representation requires the radius and the center.
- The weights have strong influence to the parametrization; their improper application can affect the surface construction.
- Surface to surface intersection perform better when using traditional forms than when using NURBS. For example, it is very difficult to handle simple 'touching' or 'overlapping' of NURBS surfaces.
- The basic algorithms, such as inverse point mapping, are subject to numerical instability

It is important to be underlined that the problems listed before are not only typical for NURBS, but also other free-form schemes, such as those of Bezier, Coons, and Gordon, have the same problems.

2.3 Geometric modeling strategies

Geometric Modeling consists in defining realistic visualization of shapes in a computer graphic environment. In the background, the geometric modeling deals with mathematical methods for shape representation. The concept of geometric modeling has been initially introduced in the mid-1970s with the development of computer systems.

When designing complex product shapes that are not fully known in advance or do not exist yet, the usual geometric modeling procedure is interactive and involves the following steps (Patrikalakis, 2003):

- *Construction of the shape based on design requirements*
- *Evaluating the overall shape of the object*
- *Improving the shape by modifying it until a satisfactory design is achieved*

The aim of the Geometric modeling is to provide a complete, flexible and real representation of the object that is even not fully known, so that the shape of the object can be:

- Realistic visualized (rendering)
- Flexible for manipulation (modification)
- Easily convert to a model that can be analyzed computationally
- Successfully manufactured and tested

2.3.1 Surface Modeling in product design

Before computer-aided surface modeling existed, automobile profiles had to be designed by making a clay model of car bodywork and, by trial and error methods, this model was shaped until aesthetically and functionally satisfactory. The geometry of the car body panels was then determined by physical measurement of the clay and stored in form of 2D technical drawings and documentations. The development of computer hardware and graphic introduces the possibility for developing computer-aided geometric modeling systems. The development of the geometric modeling provides a fast and flexible platform for storing the geometry of a physical model that can be later used for visualization or manipulation. The basic geometric modeling strategies included in today's computer-aided design systems are: wireframe, surfaces and solid modeling. Wireframe modeling represents objects by edge curves and vertices on the surface of the object. The wireframe model represents the edges of the object in Euclidian 3D space, which is much better than projection of the edges onto a plane. The 3D wireframe model are nonetheless limited in that they cannot provide an unambiguous representation of the object and do not contain any information about the interior of the object (spatial addressability). Surface modeling defines objects

with greater mathematical integrity as it models the surfaces to give more definitive spatial boundaries to the design. Unlike the wireframe modeling, the surface modeling enables visualizing more complex surfaces without producing a physical object (e.g. industry). The benefit of surface modeling using a computer is the possibility of having both the aesthetics and 3D geometry of the surfaces shape defined in one process. The surface models are types of 3D geometric models of a shape with no thickness. Clear differentiation between surface models and tick models has to be made because the latter have mass property, possibility of collision detection, and possibility for generation of finite element mesh. Solid modeling is the most advanced geometric modeling method for modeling in three dimensions. The complete geometric data representation of an object is ability to classify points in space relative to the object in terms if the point is inside, outside or on the object. The solid models are capable of handling spatial addressability as well as verifying that the model is well formed. The latter means that these solid models cannot verify whether two objects occupy the same space. Since the application domain of this work is the industrial design of products, the surface models are considered as a medium for transferring the aesthetic properties and the emotional affection to the customers. The surface models can be created using two different groups of techniques divided in: free-form and subdivision surfaces, on the on hand, and B-Rep and declarative modeling, on the other hand. Therefore, these surface modeling techniques are listed in the following:

2.3.1.1   Free-form surface

Free-form surfaces are widely used in all engineering design domains where the shape of the final product is more important from costumer point of view in terms of aesthetics. Therefore, the free-form surfaces modeling is also called the art of the engineering free-form surfaces. In general, there are two main methods for creating free-form surfaces: Lofting and Sweeping. Lofting represents an exact interpolation of a surface by using set of curves. Sweeping operation consists of creating a free-form surface by sweeping a profile curve along a trajectory curve in space. The most common types of spline that are using in designing free-form surface are the NURBS (Section 2.2.2). This is very important in cases where there is intersection of splines and analytic surfaces. The free-form surfaces are defined by control polygon and their modification depends more on the way how the free-form surfaces is created. In case the surface is created using either loft or sweep, the modification of the surfaces is possible by modifying the profile curves (loft and sweep), or the trajectory curves (sweep). An alternative way of modifying of all free-form surfaces, regardless the way of creation, is possible by moving the control points (Figure 2.11). Such modification of free-form surfaces is very tedious, less intuitive and limits the creativity of the designers. The modification of free-form shapes, performed by manipulation low-level geometric entities (points and curves), does not provide certainty that the final shape will satisfy the aesthetic

requirements. Additionally, this character of the modification of free-form shapes discourages the non-designers to use them in designing their own products.
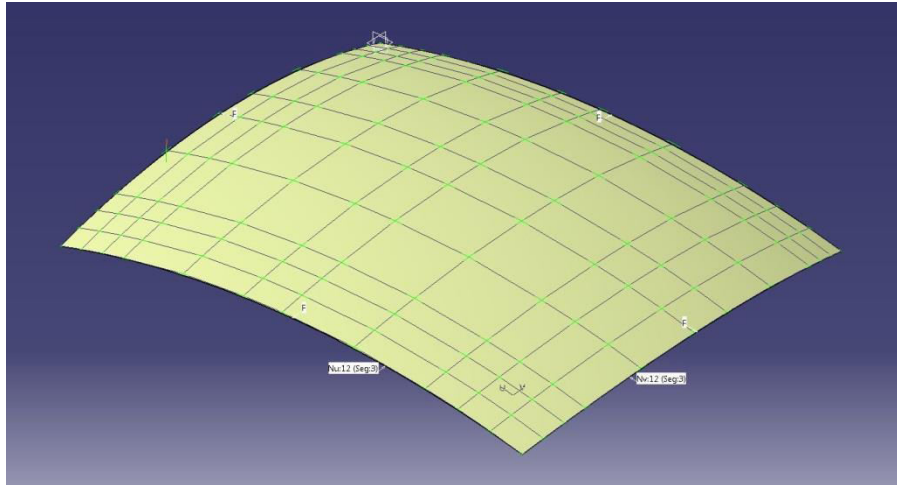
Figure 2.11: Free-form surface

2.3.1.2 Subdivision surfaces

Subdivision surfaces have become popular in animation and are gaining interest also in a computer aided design application. Usually, the modeling of complex and smooth surfaces such as those appeared in human character animation is by using trimmed patches of NURBS surfaces. Trimmed NURBS are wide spread in surface modeling because they are available in almost all existing commercial system but, according to DeRose et al. (DeRose, Kass, & Truong, 1998), they do have at least two major drawbacks: 1. trimming is a very time-consuming activity which tends to numerical error and 2. it is very difficult to preserve the smoothness at the joined edge of the surface patches during the animation of the model. Subdivision surfaces are capable to solve all problems related to the appearance of smooth surfaces including the two major difficulties of the trimmed NURBS surfaces. In other words, subdivision surfaces do not trim surfaces, and the smoothness of the model is automatically obtained even during the animation of the model (Figure 2.12).

On the other hand, the use of subdivision surfaces introduces new challenges throughout the production process, from modeling and animation to rendering. Regarding the modeling using subdivision surfaces discharges the designer from considering the topological restrictions that are mandatory for NURBS modelers, but at the same time they limit the use of manipulation (designing and modification) tools that have been developed before in order to add or modify features such as variable radius (e.g. hole or fillets) to NURBS models.

28
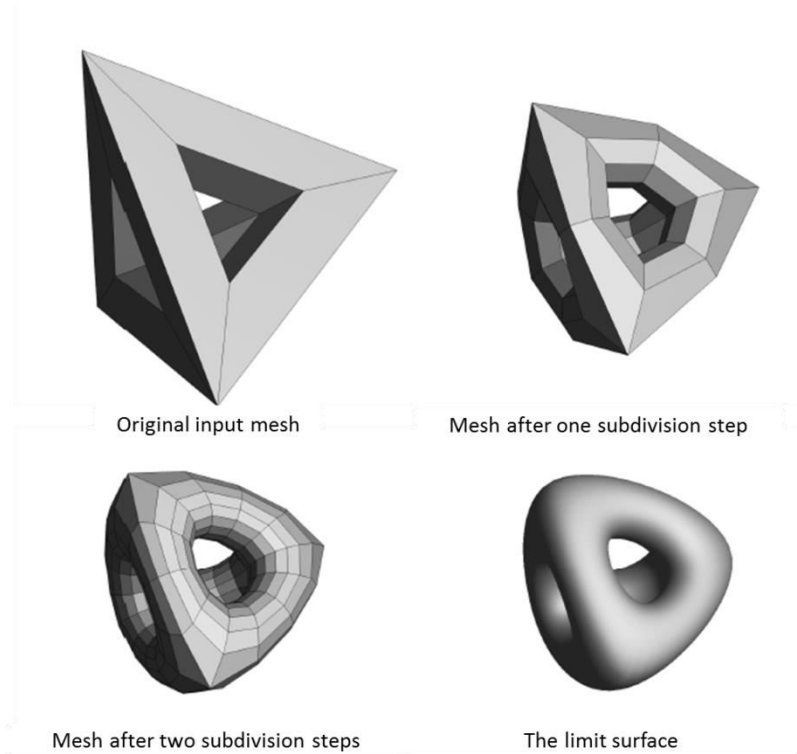
Figure 2.12: Catmull-Clark subdivision scheme (DeRose, Kass, & Truong, 1998)

Another application of subdivision is in multi-resolution of shapes (surfaces) where the representation of the shape is based on the continuous regular mesh and it is required by the need of representing of the shape in different level of details. In other words, a typical application of multi-resolution property is the real time rendering of large scenes, where an object near the viewer is represented in detail to increase the visual quality rendering, while an object farther away is represented at a coarser resolution to save rendering time. In addition to this, multi-resolution property is also convenient in case where a modification of the shape is required. For instance, the user can use the coarse resolution of the shape so as to be able to make a large scale modification or finer resolution in order to work on modification of details (Catalano, Ivrissimthis, & Nasri, 2007).

2.3.1.3 Boundary Representation (B-Rep)

Various representation modeling paradigms have been developed to create solid models of real objects. Due to the importance of solid modeling, different solid modeling approaches have been developed, divided in several groups: Constructive Solid Geometry (CSG), Boundary Representation (B-rep), cell decomposition, free-form parametric solids, swept volumes, and partial differential equations (You, Chang, Yang, & Zhang, 2011), (Hsu, 2010). There are two well-established and most popular modeling techniques for represen-

tation of solids in CAD systems (Hoffmann, Geometric and Solid Modeling: An Introduction, 1989), (Ault, 1999), (Hoffmann & Vanecek, Fundamental Techniques for Geometric and Solid Modeling, 1991): Constructive Solid Geometry (CSG) and Boundary Representation (B-Rep). Since the B-Rep can represent closed objects with free-form surfaces, this representation model will be presented in the following.

Boundary representation (B-Rep) is one of the most popular and widely used representation schemas that can store all necessary information related to the object boundaries (vertices, edges, and faces). A B-Rep of a solid defines the solid as a set of oriented and connected faces forming a closed surface. The orientation of the bounding faces needs to easily decide easily between solid material and empty space (Figure 2.13). Thus, a boundary representation allows determining whether a given point is inside, outside, or on the boundary of a volume. The boundary of an object consists of vertices, edges, and faces. For each of these entities (vertex, edge, and face), the geometric part of the representation fixes the shape and/or location in space, and the topological part record the adjacencies. The combination of topological and geometrical information is a *Boundary Representation* (*B-Rep*) of a solid (Hoffmann & Vanecek, 1991).

The B-rep is widely adopted in computer graphics because it represents explicitly the information of the object enclosing surfaces (geometry and topology). The most important characteristic of the B-Rep modeling is that it is very convenient to construct solid model of irregular and complex shapes that are difficult to build using primitives (CSG). The B-rep modeling enables the representation of free-form shape that is extensively used in both aesthetic and engineering design. This type of representation has been used for modeling objects that will be further used in the investigation of the possibility for mapping aesthetics to surface shapes.
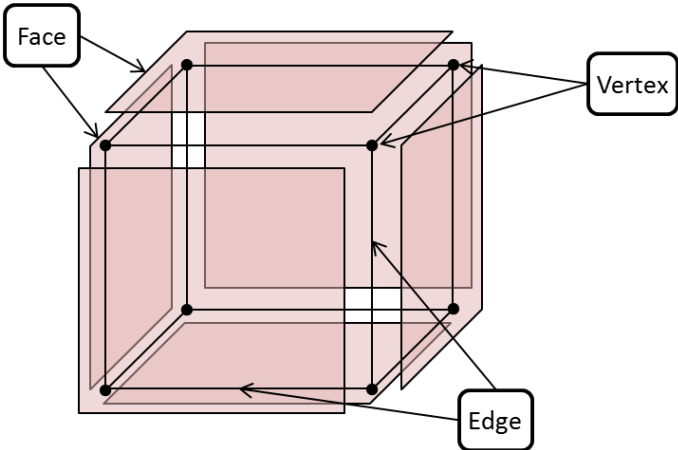

Figure 2.13: Boundary Representation (B-Rep)

2.3.2 Procedural design process

The development of the Geometric modeling in computer graphics followed the requirements for designing significantly complex geometric models, which are also enriched with semantics. Earlier geometric models, such as polygonal model, NURBS, points, and so on, do not contain enough information, so that the models can be enriched with meanings. Higher-level modeling techniques have to provide an abstraction of the model, distinguishing classes of the objects, and to allow high-level manipulation of models (Ebert, Musgrave, Darwyn, Perlin, & Worley, 2003). Many of these advanced geometric modeling techniques are inherently procedural. Two major approaches exist to geometric modeling mechanisms, namely *rule-based* and *imperative* modelers (Bardis, Makris, Golfinopoulos, Miaoulis, & Plemenos, 2011). In *rule-based* modeler, the user designs the geometric model with the aid of a set of rules (Muller, Wonka, Haegler, Ulmer, & Van Gool, 2006), whereas in *imperative* modelers, the user defines the construction of the models through procedures following parameter values (Meiden, Hilderick, & Bronsvoort, 2007). The latter approach of modeling is the case with the majority of commercial CAD modeler, utilizing a history-based model in order to describe complex elements. The first approach (rule-based) is considered as declarative modeling where the abstraction is composed using intuitive terms provided by design environment. The second approach is the procedural modeling which aims at automatic creation of large scenes in computer graphics through algorithms or sets of rules (Ganster & Klein, 2007).

The most common application of the procedural approach is in the creation of complex shapes or animation of phenomena too difficult to specify them explicitly. One type of the procedural modeling approach is the definition of shape grammars. The Shape Grammars (SG) has been introduced for the first time more than three decades ago by Stiny and Gips. In the recent years, SGs have been widely utilized in the CAD systems for reproducing large number of aesthetically consistent and novel designs (Stiny & Gips, 1971). The procedural modeling approach based on shape grammars depends on the identification of high quality grammar rules of the shape with sufficient accuracy, but in the meanwhile preserving a high degree of generality, in order for the system to create large variety of forms with numerous variations. The application of the rules generates designs, and the rules themselves are descriptors of the generated designs. The generation of variety of design shapes using shape grammars rules is a part of a design strategy called generative design (GD). The Shape Grammars (SG) together with L-systems (LS), cellular automata (CA), generic algorithms (GA), and swarm intelligence (SI) represent the five most commonly used Generative Design techniques (Singh & Gu, 2012),which are used as basis for developing the most existing GD systems.

The procedural design process is a type of design where the shape of the desired object is obtained by guiding the designer to follow previously defined designing steps. Ebert et

al. claimed that procedural shape generation techniques provide great flexibility, data abstraction, and relief from specification of detailed shapes (Ebert, Advanced Geometric Modelling, 1997). Procedural techniques provide the shape to be manipulated by using high-level manipulation parameters. For instance, the user modifies the shape of a glyph from a more directorial, indirect aspect, where he/she is free from the entire explicit specification of detailed shapes. (Ebert, et al., 2000).

The term procedural modeling has wider meaning (e.g. geometric, system or process modeling), depending on which application domain is used. In general, the procedural modeling is a semiautomatic output generation using a rules or procedure. The possibility to generate a wide range of detailed data with minimum human involvement, opened perspective for applying the procedural modeling in the field of virtual environments, increasingly used in movies, games, and simulations (Smelik, Tutenel, Bidarra, & Benes, 2014).

The advanced modeling techniques for designing surfaces are divided in three groups: 1: NURBS-based surfaces, 2. Linear surface manipulation (extrude and revolving), and 3. sweeping/lofting free-form surfaces. In many contemporary modeling systems, sweeping proves to be a practical, simple, and very efficient method for modeling various free-form surface shapes. The basic idea is to choose some geometrical object (generator, cross-section, profile), which is then swept along a curve (trajectory, guide) in space (Figure 2.14).
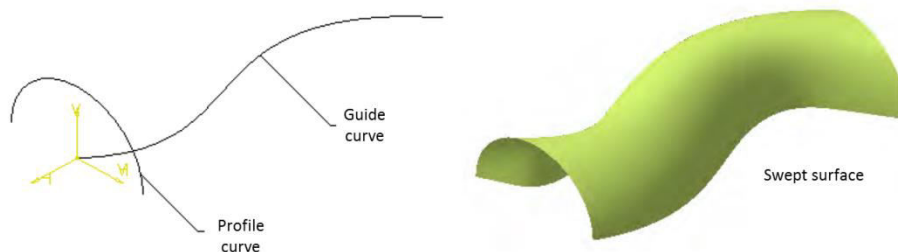
Figure 2.14: Sweeping profile curve along guide curve generation swept surface

The result of such evaluation, consisting of motion through space and intrinsic deformation, is a sweep object. The profile curve can be guided by one curve or two. The swept object type is determined by the choice of profile and guide curve and relationships among them (Marhl, Guid, Oblonšek, & Horvat, 1996). Therefore, the procedure of definition of the swept surface consists of two vital steps:

1. Selection of profile curve
2. Selection of guiding curves (one, two, or three)

However, beside the easiness of creation of swept surfaces, their modifications still remain less intuitive and depend on modifying the shape of the profile and guide curves. The manipulation of the geometric quantities in order to modify the shape of the surface requires enhanced skills of the designers and foreknowledge of geometry. In addition, the modification of the shape is exceptionally difficult in case we want to modify the shape, but preserve its borders. This geometry-driven modification of shapes is not only far away less intuitive, but also does not ensure that the reached shape will satisfy the aesthetic constraints and requirements. Therefore, it would be much better if the modification of the shape is aesthetic (emotion)-driven by the designers, whereas the manipulation of the geometry is hidden in the modification tools.

2.3.3   Needs for intuitive modification of geometric models

In the field of industrial design, objects can be exposed to functional and aesthetic constraints such as shape aerodynamics, smoothness, and so on. In case where CAD software is considered, the deformation of the shape has to satisfy some geometric constraints, usually related to the functional requirements. Guillet and al. suggest that in order to significantly reduce the number of optimization parameters, the optimization of the shape of free-form surfaces mostly requests the use of a reduced set of global parameters for controlling of the surface deformation (Guillet & Léon, 1998). The deformation techniques of surfaces are devoted to the modification of the shape of objects.

The geometric representations (e.g. NURBS, B-Spline, Bézier, etc.) used in designing activities are characterized by a large number of control points. Further, the overall shape of an object must be decomposed into a set of NURBS surface patches to get a complete shape representation. Thus, without an appropriate three-dimensional modification tools for modifying the entire shape, the surface deformation leads the designer to tedious manipulation of a single patch (i.e. displacement of numerous control points). In order to obtain modification of the entire shape, the manipulation of a single patch has to be repeated for all patches of the shape. The basic aim of these deformation tools is to provide the user with easy and intuitive control of the surface shape. Since the geometry-driven deformation requires foreknowledge of geometry and great skills in geometry manipulation, it is considered tedious and less intuitive. Therefore, it is very important, at the end of the deformation activity, that the final shape satisfies certain aesthetic properties as well – aesthetic-driven deformation.

2.4   Synthesis and Conclusion

In this chapter, the basic geometric representation methods have been presented. The most widely applied geometric representation methods in geometric modeling are *parametric*, *implicit* and *explicit* methods. After comparing them, it appears that the parametric method is the most suitable application tool for geometric modeling in product design processes and has been adopted by all commercial CAD systems. The parametric representation method consists of Bézier, B-Spline and NURBS curves and surfaces. Further in this chapter, the properties of the basic parametric representation have been presented. The Bézier curve (surface) is the simplest representation and, at the same time, the first formalization of curve (surface) that provides the possibility to precisely describe, and the opportunity for more intuitive manipulation of the shape of complex objects. The drawbacks of the Bézier curve (surface) are overcome by the formalization of the B-Splines, but they (B-Spline) are still unable to represent analytic shapes. The NURBS model is generalization of the B-spline model and is able to represent the analytical and complex free-form shapes.

The NURBS have become a standard representation method used in the geometric modeling strategies that are part of all CAD systems. This chapter introduces the basic surface representation techniques, such as: free-form surfaces, subdivision surfaces and Boundary Representation (B-Rep). The aesthetic character and impression of the product is transferred to the customers through its surface shapes. Therefore, the surface models are the most appropriate geometric models for mapping aesthetic properties. Actually, the surface models are those that represent objects for which aesthetics are important. Regarding the geometric modeling mechanisms, there are two major approaches: procedural and declarative modeling process. The former approach is used in almost all commercial CAD systems. The main characteristic of the procedural modeling process is the need for defining many low-level constructive details that are, further on, used in the procedure for creating the overall shape. The definition of low-level constructive elements restricts the creative activity in the design process, which is crucial in designing appealing products. This, not only restricts the creativity of the designers, but also disables greater inclusion of non-designers people in designing appealing shapes. The role of the non-designer people has been even more emphasized by the development of new manufacturing technologies, such as 3D printers (Additive Manufacturing equipment) where non-designers can design and produce their own products. On the contrary, the declarative modeling is based on using more abstract and intuitive notions. The main difference regarding the procedural approach is the definition where it is more important what shape is rather than how it is built. Therefore, the declarative modeling approach tends to overcome these restrictions and to propose more intuitive and target-oriented modification tools. Emphasizing the aesthetic aspect and designing appealing products introduce the need for developing intuitive and application-oriented modification tools for direct shape generation by non-professional users.

Aesthetic-oriented free-form shape description and design

This chapter presents the bibliographic works related to the basics of declarative design and aesthetic properties of shapes. It begins by introducing the needs for developing high-level tools for the modification of geometric models (Section 3.1). This section is organized in three subsections. The first part is dedicated to the declarative design process (Section 3.1.1) presenting it as a modeling approach, positioned on the conceptual and semantic modeling levels. The second part (section 3.1.2) presents the target-driven designing concept that implements the declarative modeling approach. The final part (Section 3.1.3) introduces the aesthetic-oriented description and manipulation of 3D free-form shapes in industrial design as part of target-driven design. Further in this chapter (Section 3.2), the aesthetic properties of curves proposed by the FIORES II Project are presented. In the next section, the aesthetic properties of surfaces (3.3) are discussed. The last section (3.4) gives a synthesis of this chapter.

3 Aesthetic-oriented free-form shape description and design

3.1 Toward high-level modification of geometric models

There is a rich variety of geometric representations for describing and manipulating complex geometric shapes and especially, surface models. As explained in the previous chapter, the B-Rep representation is the most commonly used in commercial CAS system (Hoffmann, 1989). The key factor that contributes to this is the ability of a B-Rep to use NURBS allowing local shape modification (properties of NURBS curves and surfaces), which is used by a wide range of methods integrated in CAD systems. The most common use of the B-Rep is in the mechanical engineering field for representing regular shapes of mechanical parts. In this domain, the so-called features can be repeated many times in different objects (e.g. holes, shafts, screws, and so on). In order to avoid the repetition of the same part, the feature-based designing concept has been introduced.

3.1.1 Feature-based approaches

The concept of feature-based design is based on defining features that are high-level entities, and designers can use them to construct the final shape of the mechanical part. The

main benefit of the implementation of feature-based designing concept is that designers do not manipulate directly the surfaces (e.g. the control points), but the parameterized features which can be parameterized by many numerical parameters such as length, width, height, depth, and so on. Hence, the designing process has been transformed into a construction process where the final shape of the mechanical part is constructed by combining different features preserving the relations between all construction entities (features). Moreover, the modification of the shape is reduced to simply changing values of the feature parameters that result with modifying the entire shape of the part (Pernot, Quao, & Veron, 2007). Unfortunately, the feature-based concept in mechanical design does not support definition of shapes using free-form surfaces in the entire design process. The free-form surface design process is in the scope of the thesis. Actually, the results of this thesis can be used to improve current free-form surface modeling tools whose use still requires a deep knowledge of the underlying mathematical models.

Analyzing the advantages of feature-based design, it is evident that the manipulation of features is more intuitive and reduced to varying the value of few numerical parameters than manipulating directly the surfaces. Contrary to this, the manipulation of free-form surfaces (Bézier, B-splines and NURBS) is performed by direct manipulation of surfaces, and requires foreknowledge of geometry description, and great skills in geometry manipulation, which makes it tedious and less intuitive. The sublimation of both, the advantage of the feature-based designing and the requirement for the free-form shapes in terms of manipulation, drive to the need for defining higher-level free-form modification entities. Since, analytic surfaces cannot represent the complex shapes widely used in industrial (aesthetic and engineering) design, commonly defined by several NURBS patches connected together; the concept of form features has been extended to the free-form domain (Pernot, 2004). The objectives of defining high-level free-form entities are to have more *application oriented elements* than the mathematical low-level constructive elements (points, curves). The free-form entities are easy to be manipulated by using tools, which can play a role of intent-driven modifiers of free-form features (FFF). The introduction of the free-form features concept opens perspectives for creating tools for high-level modification of geometric models.

Fontana et al. (Fontana, Giannini, & Meirana, 1999) have proposed a formal classification of detail free-form features (Figure 3.1). The proposed classification consists of two main classes of features that correspond to features obtained by deformation (δ-FFF), and features obtained by elimination (τ-FFF). The classification of deformation features (δ-FFF) is based on the topological and morphological properties associated to the deformation function used to create the features.

The morphological property distinguishes the intrusion from the extrusion, while the topological property distinguishes the border, the internal, and the channel features. The classification of features obtained by elimination (τ-FFF) distinguishes different classes regarding the finishing operation, either a sharp or a finished cut (Figure 3.1).
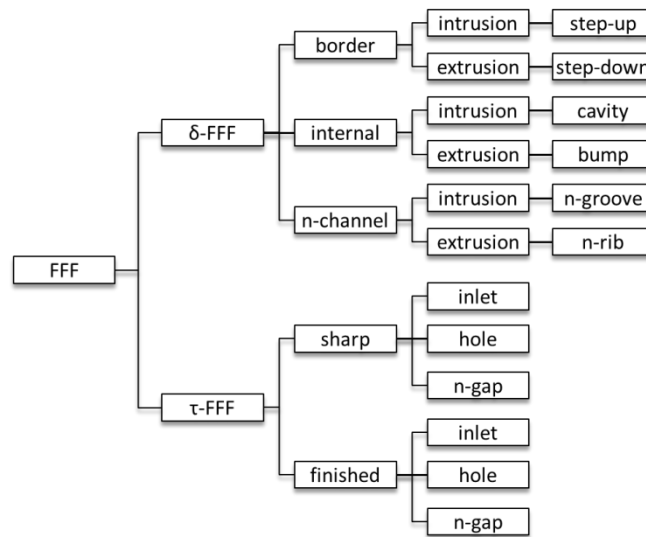
The paper (Fontana, Giannini, & Meirana, 2000) presents a first extension of the free-form features concept to aesthetic design. Features are high-level modeling entities that offer a valid support in storing the design intent and making easier model modifications. Analyzing the styling activity, it can be concluded that the creation of a shape is performed by following two steps: designing the overall shape of the product and enriching it by adding local details. Therefore, depending on the phase of the design process, two categories of free-form features can be identified (Fontana, Giannini, & Meirana, 1999):

- Structural features, designed in the initial modeling phase. They represent the main product parts that are used to give the overall shape of the products, thus having an emphasized aesthetic impact.
- Detail features, designed in the later modeling phases. After defining the overall character of the product by the structural features, the detailed features are added to enforce the visual effects of important shape elements.

3.1.2 Declarative design process

All available geometric modelers make it possible to construct complex shapes in a more or less intuitive manner. Nevertheless, these geometric modelers restrict the designers in terms of description, using a set of point coordinates or basic geometric primitives, making the designing process very complex and tedious. The role of these modelers is to convert input data of the desired object into an internal numerical model through procedures, so called imperative or procedural modeling (Meiden, Hilderick, & Bronsvoort, 2007). The intention to avoid the use of low-level geometric quantities (points and curves) for defining the object motivates many researches to develop modelers that will help us to design ob-

jects using more abstract notions, based on properties (geometric, topological and physical) and constraints. This is why the concept of declarative modeling has been introduced (Lucas, Martin, Philippe, & Plémenos, 1990). The definition of more abstract notion for describing the surfaces opens perspectives for defining high-level modification tools for modifying the free-form shapes. The role of the computer is to provide a flexible environment for designers to create shapes and, using suitable tools, to modify the shape satisfying their requirements.

Geometric declarative modeling tends to design a desired object by expressing its properties. The internal computations of all numerical values (e.g. control points), necessary for the definition of the object, are performed by the modeler and are hidden from the users. The designed object has to satisfy geometric properties or functional constraints. Daniel et al. (Daniel & Lucas, 1997) suggested that declarative modelers must have at its disposition tools for describing, generating and understanding the shape:

3.1.2.1   Description tools

The basic idea consists in using a set of necessary and sufficient condition to describe a set of objects completely. The main difficulty is to determine whether a precise vocabulary is adapted to a given application field. Specific vocabularies are shared by different applications. The study of vocabulary can be summarized as follows: how to describe a curve/surface without giving a list of coordinates? This does not mean that a set of words and/or an associated syntax can be easily deduced. The relevant vocabulary can be divided into three categories (Daniel & Lucas, 1997):

- ❖ **Mathematical vocabulary,** which is universal and undisputable. Shades of meaning cannot be introduced. *Concave, convex, inflexion, curvature, cups*, … belong to this category
- ❖ **Qualitative vocabulary,** which allows shades of meaning, but is subjective and sometimes differently understood. Words such as *flat, round, slender, …* belong to this category
- ❖ **Quantifiers** such as *too much, little, much, more, less, very …* enrich the description and allow variations to occur.

It can be pointed out that stating such properties does not exclude the need for very accurate modifications. Moving points directly, and thus their coordinates, seem unavoidable. The experiences gathered by the authors (Daniel & Lucas, 1997) confirm that it may be more difficult to apply a precise modification by giving a set of properties, than manipulate directly point coordinates (using a more or less automatic process). Declarative modeling must then be considered as a powerful tool for rough sketch realization, obtained from the given properties. These sketches can evidently be the inputs of a classical modeler that then

appear as complementary to the declarative modeler, the user getting rid of the most tedious part of the design.

Giannini et al. (Giannini, Monti, & Podehl, 2006) have also identified two different vocabularies (languages) used in different activities of the design process. Namely, the authors of the paper differentiate two different languages: the marketing and the designer languages. The former language is denoted as first language, whereas the latter language is denoted as second language. The aesthetic characteristics bridge both the marketing and designer languages (Figure 3.2). The first language represents an individual description of the emotion evoked by the aesthetic character. The second language represents detailed specification of the product model during the manipulation (creation and modification), according to geometric properties and constraints. This differentiation in terms of different languages aims at describing the character specification as a bridge between the first and second language. Thus, the link between the aesthetic character and the geometry becomes a two-level mapping (level 1: mapping of the geometry to the styling terms, level 2: mapping styling terms to those expressing the emotional character).
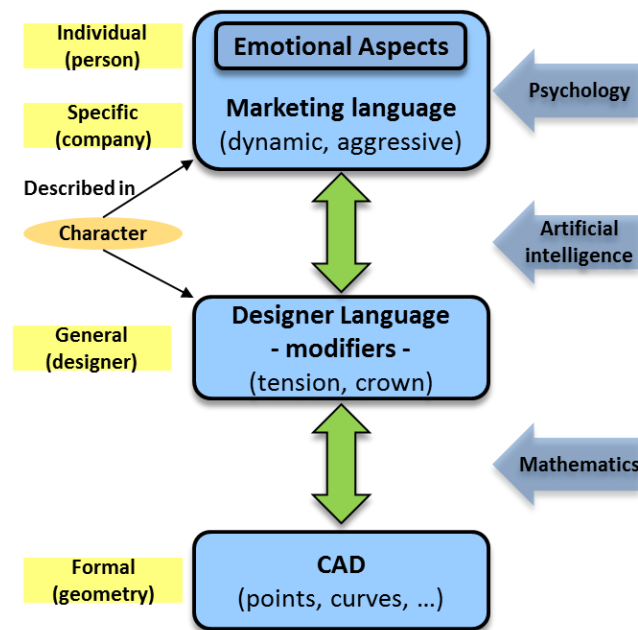
Figure 3.2: The link between the aesthetic character and the geometry is a two-level mapping [22]

3.1.2.2 Generation techniques

A problem solved through a declarative approach can have no solution, one solution, several or even an infinite number of solutions. The main problem is to transform the formal model into the geometric model. Algorithms specific to the given object, random sampling (which requires a parameterization of the objects), or methods allowing an accurate control on the produced solutions can be applied.

It seems that most of the problems encountered in the field of mechanical and industrial design can be better studied with the sampling mode, which provides one solution that can be modified by the designer. This does not restrict the number of possible solutions that can be obtained, which is necessary for a creative design. The declarative modeler has to propose, if requested, new solutions, taking into account initial properties and additional constraints that may be added by the designer.

3.1.2.3 Understanding tools

Declarative modeling requires mechanisms for a quick understanding of the created objects. This leads to the development of visualization techniques (wireframe, surface models, transparencies and so on). All the object components, a part of them or even a skeleton, will be shown. These different modes can simultaneously appear on the same image.

The aim of this part, including the description tools and generation techniques, is to present the principles of the declarative modeling and the structure of the declarative modeler.

Different modeling approaches in CAD have been developed. The distinction between the different modeling approaches is the level of abstraction used to manipulate the model, as proposed by Maculet et al. in (Maculet & Daniel, 2004):

- **Level 0**: corresponds to the manipulation of parameters (for example, a point is described using two parameters (x,y) in 2D, or three parameters (x,y,z) in 3D);
- **Level 1**: corresponds to the manipulation of basic geometric entities (points, straight lines, curves, surfaces); it corresponds to modelers for solving elementary geometric constraints (for example, distance between two points, angle between two lines, and so on);
- **Level 2**: corresponds to the manipulation of more complex objects, composed of simpler elements of level 1 (for example, groove in an area of an object); it refers to the feature-based approach which solves more complex constraints.

The second group (level 1) of modelers presents the concept of directed and undirected constraints. The third group (level 2) of modelers can deal with either geometric constraints or constraints containing non-geometrical parameter (e.g. engineering constraints can be expressed).

Michel et al. (Lucas, Martin, Philippe, & Plémenos, 1990) considered the declarative approach as an extension of the feature-based approach, beside the fact that there is no historical connection between them. The declarative approach is positioned on the level of conceptual and semantic modeling ("level 3") because of the higher level of abstraction.

Using a declarative approach offers numerous advantages. First of all, it gives more freedom to the designer modeling to let him/her describe the results he/she wishes, using a "simple" vocabulary. The vocabulary for description can be specific to a given field, which provides the opportunity of using the system even to a non-professional of geometric modeling. The aim of this approach is to provide a fast and easy way to sketch the shape of the object. Different applications with this approach have been proposed: polyhedral models (Martin & Martin, 1988), space control (Chauvat, 1994), scene modeling in image synthesis (Plamenos, 1994). In the context of industrial design, researches have been achieved on curves (Daniel & Lucas, 1997) and surfaces (La Greca, 2005). Furthermore, the shape modifiers presented in the FIORES project are following the declarative approach (FIORES, 1997).

The overall shape of a feature can be described regarding the way the geometry of the corresponding feature has been obtained, or regarding the geometric and topological entities and relations associated with dimension parameters (Hoffmann & Joan-Arinyo, 1998). The first approach is a procedural approach, whereas the second is declarative. In a procedural approach, generic features are predefined in terms of a collection of procedures, which guide the designer to build the final shape. The procedures may include methods for managing a feature (like instancing, copying and deleting a feature), and methods for specific operations on a given feature (like generating the geometry, deriving values for parameters, and validating features operations).

In the declarative approaches, the features and their properties are described declaratively. The main difference regarding the procedural approach is the definition where it is more important what the feature is, rather than how it is built. The use of constraints in order to define the features is one of the main tools of this approach (Hoffmann & Joan-Arinyo, 1997). In (Hoffmann & Joan-Arinyo, 1993), the authors proposed the E-rep declarative framework that presents an understandable differentiation between definition and construction. The constraints provide very natural and intuitive ways to describe the spatial relations between geometric entities in a feature and between features. Moreover, they provide techniques for defining the relations between geometric and technological parameters. Therefore, in declarative feature representation, constraints play a key role.

3.1.3 Target-driven design

Declarative modeling is placed on the level of conceptual and semantic modeling, where the level of abstraction is higher than the other model representations. The semantic meaning of the shapes is highly depending on the specific application field, and the vocabulary used to describe the shapes is also specific to a given field. Therefore, the declarative approach is considered as application-oriented or target-driven design. The aim of a target-driven design is to substitute the current trial and error loops of designing approaches by

direct target-driven design where the desired shape properties of the final product are parameterized. In the classical designing process, before the computer was involved, a stylist played a central role in shape definition and product design process. He/she created the clay model of the shape by using simple tools. He/she had all knowledge needed about the material and tools, he/she was the creator of the shape he/she wanted to achieve, and he/she was the only one who decided whether the results are good or not. As soon as the designing process was enriched by including different specialists and CAD systems, two problems occurred almost inevitably (Dankwort & Podehl, 2000):

- **Achieving quality**
  The direct interaction between a stylist and working material using his/her tools is no more available. The CAD model generation process will never provide the user with information on how close he/she already is regarding the desired shape.

- **Retaining the design intent**
  While working, the results are passed on from one person to the other (e.g. from stylist to CAD designer). Then, the designer tries to reach the stylist's goal, he/she never knows whether the stylist's goal (harmonic shape) has also been reached, because the intention of the latter can hardly be formulated.

The problem of preserving the design intent is difficult to solve, since the stylist´s intention has to be adjusted to the other parties involved in the process. A solution is to integrate both the knowledge of the stylists and CAD designers in a unified CAD system. The declarative modelers can be considered as potential candidates for solving this problem, but it has to be underlined that the defined vocabulary has to be extracted from the application domain (from non-professional designers). This thesis proposes a framework for mapping aesthetic properties to free-form shapes, which have been taken from non-professional people (potential customers) and not from stylists (designers).

3.1.4 Aesthetic-oriented design and modification of free-form shapes

3.1.4.1 Mapping of aesthetic properties to 3D free-form shapes

The association of aesthetic properties to 3D objects is carried out mainly through their representation by means of free-form surfaces. The Bézier, B-spline and NURBS curves and surfaces are the most flexible types of geometric representation models developed for various industrial design intents. These models are not only convenient for designing complex shapes, but they also modify them.

Since in aesthetic design the modification of the shape plays a very important role, then the free-form feature concept seems to be the best approach for solving the shortcomings related to the modification.

The process of industrial design consists of conducting many activities at the same time and involves different experts' knowledge aiming at achieving the best technological requirements while satisfying the stylist's design intention and constraints. The use of computer systems helps designers to manipulate with the product shape in different engineering phases of the design process, including the aesthetic character. The link between the shape and the aesthetic character expressed in the form of values can help specifying the shape characteristics and parameters that correspond to the design intention. In order to understand the relations between emotions and product shape from different perspectives (e.g. perceptual phycology, design and computer science-artificial intelligence (Giannini & Monti, 2002)) several researches have been conducted. Having in mind the fact that the perception of people is affected by their experience and culture, it is almost impossible to make an overall definition of the aesthetic character. In order to discover the relevant aesthetic characterization of shapes and to define which aesthetic properties the stylists consider when designing shapes, the European Project FIORES – II has been carried out (FIORES-II, 2000). General objective of the FIORES – II project was to improve the working procedures and the computer aided tools adopted by designers for modeling products shapes. According to the authors of the paper (Giannini & Monti, 2002), the new modeling tool should help CAD designers to easily create a model containing specific emotional characteristics, with respect to the stylist's design intention and to preserve them in the engineering optimization phase. This requires tools able to preserve the aesthetic properties, even if modifications of the model are required. To achieve the general objectives, the project identified the following intermediate results (Giannini & Monti, 2002):

- Identification of a vocabulary for the aesthetic design
- Mapping of styling characters on the geometric entities and properties entirely describable by measurable parameters
- Defining methods and rules to extract the aesthetic properties of the shape
- Defining methods for optimization of the design considering aesthetic and geometric engineering requirements

Being able to establish relationships between geometrical entities of a product shape and its aesthetic characteristics is the key innovation approach for developing more sophisticated modeling tools. By mathematical quantification of the specific shape characteristics, the CAD designers would be able to design directly the shape, preserving the stylist intention without trial and error looping.

Regarding the first objective of the FIORES – II project, the following list of terms that have been selected by the designers as being the most used for shape evaluation and modification is given:

- Straightness
- Acceleration
- Crown
- Convexity / Concavity
- S-Shaped
- Softness/Sharpness
- Hollow
- Tension
- Lead-in

References (Giannini, Monti, & Podehl, 2006) and (Giannini & Monti, 2010) present the identified terms and developed measurements for styling properties, together with CAD tools based on them. This list of terms should not be considered as a lexicon for styling. It is not complete and all the stylists do not use all the presented concepts. Nevertheless, all stylists and designers agreed that the list can be considered relevant, despite the fact they all come from four different European countries and work in different styling applications. In the next section (3.2), a more detailed presentation of the identified terms is introduced, together with an explicit mathematical quantification.

3.1.4.2   Defining Aesthetic Curves and Surfaces

Designing aesthetic appealing models is considered to be strongly integrated in future generation of CAD systems. A way to succeed is to introduce the concept of Aesthetic Curves and Surfaces. Many researches have been conducted, aiming at defining aesthetic curves and surfaces which can be found in many artificial and natural objects. The term 'Aesthetic Curve', for the first time was introduced by Harada et al (Harada, Mori, & Sugiyama, 1995) denoting curves with a Logarithmic Distribution Diagram of Curvature (LDDC), which is a straight line under angle. This concept is based on the definition of methods for controlling the curvature distributions. Later, in 2005, Miura et al. made a significant advancement by proposing a new category of curves, the so-called Log-Aesthetic Curves (Miura, Sone, Yamashita, & Kaneko, 2005). These curves are considered an extension of Harada et al.'s work on Logarithmic Distribution Diagram of Curvature (LDDC) by deriving a mathematical equation of curves with a Logarithmic-Curvature Graph (LCG). According to the paper (Yoshida & Saito, 2007), the Aesthetic Curves have the following properties:

- Many curves in artificial and natural objects can be represented by Aesthetic Curves
- The Clothoids, Logarithmic spirals, circular involutes and circles are considered as special cases of Aesthetic Curves
- The curvature is linear and monotone
- One parameter is used to control the curvature variation (α)
- An Aesthetic Curve segment can be generated using three control points and a parameter α.

The notion Aesthetic Curve has been used to mean beautiful curves. Farin (Farin, 1996) has pointed out some common characters of beautiful curves that are present in their curvature distribution. For instance, if the changes of the curvature are constant (i.e. the second derivative of the curve is monotone, increasing or decreasing), then this curve is considered beautiful. Contrary to this, if the curvature changes are not constant (i.e. the second derivative is not monotonic), then the curve is rarely beautiful (Kanaya, Nakano, & Sato, 2007). The authors of (Miura & Rudrusamy, 2014) introduce the classification on fair curves and surfaces continuing to the formalization of Aesthetic Curves and Surfaces. The term fairness is a well-known term used to describe the quality of curves and surfaces, and this group consists of previously exposed basic geometric representation methods – Bézier, B-spline and NURBS. Further, six different types of Aesthetic Curves, Logarithmic spiral, Clothoid, Quaternion IC, GCS, Log-Aesthetic and GLAC curves have been introduced, including their mathematical interpretation. It discusses in more details the properties of Log-Aesthetic (LA) curves and surfaces and their application in industrial design. LA curves are considered to have very convenient properties for practical use and open perspectives to become standard curves for aesthetic design. Regarding LA surfaces, this paper (Miura & Rudrusamy, 2014), discusses on how to extend the LA curves into surfaces.

The concept of defining Aesthetic Curves and Surfaces is very good, but there are many shortcomings in terms of their use in designing more complex shapes that can appear in a real product. Since the Aesthetic Curves and Surfaces are curvature-based models, they reduce the possibility of intuitive manipulations by non-professional designers. Therefore, the concept of mapping aesthetic properties to 3D free-form shapes has been adopted. The next section introduces the basic aesthetic properties of curves defined in the FIORES II project.

3.2   Aesthetic properties of curves

The aesthetic shapes of the product are represented by using free-form surfaces. The main geometric quantities used to capture and preserve the aesthetic character of the surfaces are the curves (2D and 3D), which are the basic elements for defining the shapes of industrial products. Many styling features and properties seem to depend directly on curvature. However, these curves produce complex curvature function that may undermine the formulation of aesthetic shapes. The curvature-dependent function of aesthetic properties proposed in FIORES came up with reasonable results, but did not always suit the designers thinking (Podehl, 2002). In contrast to this, stylists judge the curves depending on their personal impression and the context they are used in, which makes it difficult or even impossible to find one similarity function to fit all needs.

Therefore, it is necessary to find meaningful measures for properties' evaluation allowing the control of the shape by simply changing the value of the property measure. By con-

trolling the value of the property measure, it is possible to change the geometric properties related to the given property and hence, to control the shape.

3.2.1 Straightness

Used as an engineering term, a straight line is the shortest connection between two points, which is a linear curve (in design, line and curve are mainly used synonymously). A linear curve has zero curvature. This leads to a design definition of the term straight: A straight line is a curve with infinite radius. While in engineering, a curve is either straight or not (apart some tolerance), for a stylist, a curve can be more or less straight, depending on how visually it differs from the line segment which is somehow related to the dimension of the overall curvature radius in relation to the curve length. The bigger the radius, the straighter the curve is.
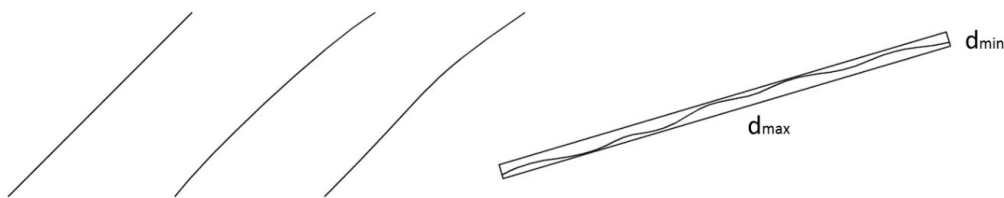
Figure 3.3: Straight curves: engineering, styling, s-shaped and noisy (left to right)

Even curves having inflection points and consequently variable radius can appear somehow straight. The FIORES – II project proposed the following formulation for the straightness measure:

$$straightness = 1 - d_{min}/d_{max} \qquad (3.1)$$

Where d_{min} and d_{max} indicate the height and the width of the minimum bounding rectangle. Actually, the bounding rectangle does not provide any supplementary information concerning the character of the curve, i.e. how it behaves, the derivations or oscillations. In addition, totally different curves can have the same bounding rectangle.

Because of previously mentioned drawbacks, the work described in (Giannini, Montani, Monti, & Pernot, 2011) presents a definition and implementation of semantic operators for curve deformation based on a shape characterization that is specific to the industrial design context using a revised measure for the straightness.

$$NS = non\text{-}straightness = \frac{C \cdot A \cdot L}{l^2} \qquad (3.2)$$

$$non\text{-}straightness \in [0, \infty)$$

where, *C* is the integral of the absolute value of the curvature, *A* is the value of the area between the curve and the line that joints the two extremes of the curve, *L* is the length of the curve, and *l* is the length of the cord between the two end points of the curve.

$$S = straightness = \frac{1}{1+(non-straightness)}$$ (3.3)

straightness (S) $\in$ (0, 1]

This measure of straightness (eq. *S = straightness* = $\frac{1}{1+(non\text{-}straightness)}$ (3.3) represents the character of the curves better than the initial one (eq. *straightness = 1 − d_{min}/d_{max}* (3.1). The equations for the geometric quantities (*C, A, L, l*) and the initial qualitative classification of the straightness property according to property range values has been further detailed in Chapter 5 (Sections 5.3 and 5.4).

3.2.2 Acceleration

Acceleration is a term used to describe curves with rising curvature. A curve without any acceleration is a straight line or a circle. Fast or slow acceleration means that the curvature increases fast/slowly. If a curve changes curvature slowly, it may show no acceleration at all. Acceleration is sensitive to the orientation of the curve on which it applies. There is no unique definition at which point a curve starts to accelerate, but acceleration always starts in a rather flat area and evolves into a high-curvature region. Moreover, stylists say that symmetrical curves have no acceleration. Considering this styling property as a local property, it may be defined a measure for acceleration in a region by the ratio of curvature difference (D_k) and the arc length difference (D_l).

$$acceleration = D_k / D_l$$ (3.4)

The degree of acceleration in one point can then just be given by the rate of curvature change (the third derivative of the curve parameterized in arc length). The higher the change, the greater the acceleration is. Considering the whole curve, the acceleration is related to how much the variation of the tangent to the curve is balanced along it. Thus, a global measure for it has to take into account that when the 'majority' of the tangent variation is close to one of the curve extremities, the curve is said 'to accelerate around' this extremity; the closer it is to the extremity, the more it is accelerated. It must be considered that the presence of several local curvature maxima do not increase the acceleration effect;

on the contrary, if the curvature maxima are distributed along the curve, this results in a not accelerated curve.

3.2.3 Convexity/Concavity

A (non-linear) curve is convex or concave if the curvature along the curve has the same sign. Whether a curve is convex or concave depends on the context in which the curve is viewed. When designers are making a curve more convex, they are moving towards the enclosing semi-circle; that is, considering the chord between the two extremes of a curve (Figure 3.4), the most convex curve on that chord is the semicircle with diameter equal to the chord. Thus, the ideal convex curve is the semicircle, or an arc of circle, if the continuity constraints at the endpoints are compatible; otherwise, it is the curve presenting the lowest variation in curvature that satisfies the given continuity constraints. Conversely, the least convex curve on that cord is the cord itself. Judging a curve more or less convex depends on several factors: above all the symmetry, the roundness, and the curvature variation. Many of these factors depend on mathematical properties that can be calculated on the curve (i.e. length, area, gravity center coordinates and momentum of inertia). The convexity measure criterion that we consider takes into account all the factors that are implicitly considered by the users, and it is obtained by measuring the distance of a vector of curve properties from the corresponding vector computed on the ideal convex curve. Since it has been experimented that the considered attributes have a different impact on the perception of convexity, and in particular the most important ones seem to be curve symmetry and roundness, then weights have to be introduced in the measure.

Figure 3.4: Changing convexity and concavity with respect to the bottom to top direction

For a measurement, one could simply define convexity and concavity as signed curviness, where the sign must be derived from the viewing perspective of the object itself, maybe by its barycenter. Another possibility is to use the signed area under the curve (limited by the line between the two curve end points) as a measure for convexity/concavity. One could even combine both ideas by using their product (Giannini & Monti, 2003):

$$convexity = signed\ area * (1 - straightness) \qquad (3.5)$$

where positive values stand for convex and negative ones for concave.

3.2.4   Other aesthetic properties of curves

3.2.4.1   Hollowness

A property very close to convexity is hollowness, which is a less technical but more subjective concept. From the engineering point of view, a curve or surface can be called hollow, if it is concave. In design, a curve or surface can look hollow by wish or by mistake, although it is not concave at all (Figure 3.5). If, for example, an almost straight curve is connected to a rather small almost true radius, the connection usually appears hollow, because there is no smooth lead-in. The observer follows with his eyes the radius evolution which would create a truly concave transition. As a consequence, a curve may appear straighter if it is more curved - or has some crown. The ancient Greek already knew about such kinds of effects and avoided long straight horizontal lines on their temples. In order not to make them look hollow, they built the horizontal parts slightly convex.

Thus, giving a measure for hollowness may be difficult. Hollow is very close to concave, but parts can appear hollow even if they are not concave at all. It seems to be necessary to involve the curve's orientation in space, which can be seen in the fact that long horizontal lines are often judged as being hollow:

$$hollowness = concavity * arc\ length * horizontality \tag{3.6}$$

where *horizontality* is the cosine of the elevation angle for the whole curve (e.g. the connection line between start and end point) with the x-axis.

Figure 3.5: If the *radius* meets a straight line (A), the transition area appears *hollow* (B, exaggerated)

3.2.4.2   Crown

Although crown sounds like being a feature only and not a property, the term is used mainly as a modifier, like in the phrase "Put on more crown". It means lifting or raising a certain part of the curve or surface, but not changing the end points or edges. In principle, one can raise every kind of curve, but "putting on crown" can only be applied to already convex curves. There are different parts of curves and surfaces which can be given more crown, and crown is always added into a certain implicitly or explicitly given direction. Parts and directions are defined by certain base lines or sections of the curve, and the crown is added perpendicular to them (see Figure 3.6):
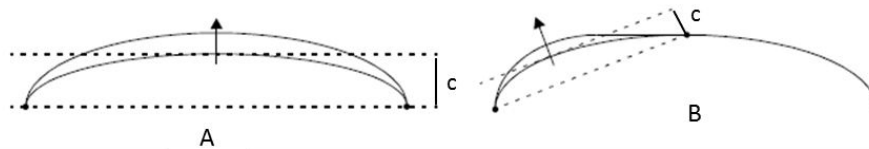
Figure 3.6: Two ways (A and B) to add and measure (c) crown with respect to different baselines

A. The baseline is the connection between start and end point of the curve. In this case the curve gets somewhat raised (mainly for symmetric curves).

B. The baseline connects two important points, such as end points, inflection points or flat points, or just points "where the acceleration starts". Then the curve gets pushed at its biggest elongation point into the direction perpendicular to the baseline (mainly for asymmetric curves).

A crown could be measured simply by the maximum elevation of a curve with respect to a chosen baseline. This baseline could be the x-axis or the connection between start and end point or – the connection between two important (user-chosen) points.

3.2.4.3   S-Shaped curves

An s-shaped curve consists of two parts of opposite curvature sign, thus obviously carrying exactly one inflection point. Or, in other words: An s-shaped curve consists of a convex and a concave part that are separated by an inflection point (see Figure 3.7). S-shaped curves are not always wanted, because the inflection point is a very outstanding curve feature, which may disturb the overall look. If an s-shaped curve is wanted, then the S should be well visible. Curves with an inflection, but a weak visibility of the S are mostly regarded as being bad.
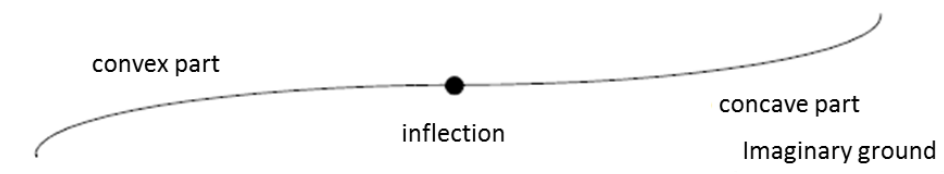
Figure 3.7: S-sharped curve

In general, s-shaped curves are not wanted in the automotive sector, since the inflection points may disturb the overall look of the curve, especially if the visibility of the 's' is weak.

To quantify an s-shaped curve, more than one measure must be taken into account:

$$biased = (2*s_{inflection}/L) - 1 \tag{3.7}$$

$$tangency = \ convexity\ (convex\ part) + concavity\ (concave\ part) \tag{3.8}$$

$$visibility = curvature*(1 - biased^2) \hspace{3cm} (3.9)$$

where *biased* indicates where the 's' (left to right) is positioned, $s_{inflection}$ is the arc length of the inflection points, *tendency* indicates which is the dominant characteristic (convex or concave) and *visibility* estimates how much the *s-sharped* is well recognized.

3.2.4.4   Tension

Tension can be understood from the physical analogy of applying tension to a steel spline. In the physical example, the tension of the curve (or the bending energy) can be found where the curvature is the highest. There are two ways to add more tension to a curve (see Figure 3.8):

Figure 3.8: Two ways for applying pressure to get more *tension*

A. Keep the end parts unchanged and flatten the middle part (clay modeling view), which is analogous to fixing the ends of the spline and applying pressure on the middle part.
B. Keep the middle part unchanged and increases the radius close to the end points, which means hold the spline in both hands and turn the hands to the outside.

The tension is related to the 'internal energy' of a curve, subject to continuity constraints at its boundaries. Several criteria such as energy (flexion, stretching, etc.) are considered, as well as some shape factors that can be compared to beam section properties (inertia, springiness). The user's axiomatic feeling is that: 'Straight lines have either no tension or an infinite one'. Therefore, the tension represents the 'energy' required to change the shape of a curve, provided it is not a straight line. Then, the higher the tension, the closer is the curve to a straight line. Many designers said that one could feel tension only if 'something happens' in the curve, which means that there is an evolution of curvature along the curve. A first attempt for measuring tension would be the ratio of curvature extremes k_{min} and k_{max}, but in order to be more global, one could set the curvature in relation to the average curvature k_{avg}:

$$tension = (k_{max} - k_{min})/k_{avg} \hspace{3cm} (3.10)$$

As "tense" curves show specific amounts of crown, one could also try to model tension by either crown in vertical direction (related to the base line), or crown in "diagonal" direction (related to a section line between two important curve points).

3.2.4.5   Lead-in

As for most of the styling work, it also helps understanding the term "lead-in" by knowing how clay modelers proceed in their work of creating an object (see also [Yamada 1993]). They start creating the main surfaces from true sweeps (constant radius templates) and connect them by blending - or a radius, as they would call it. In most cases, a constant radius connects to a curve only G1-continuous (tangent). Thus, this hard connection does not lead well into the transition. The curve or surface needs to be modified so that it smoothly leads into the radius and, thus, look harmonic. If "more lead-in" is wanted, we can do it in several ways (see Figure 3.9):
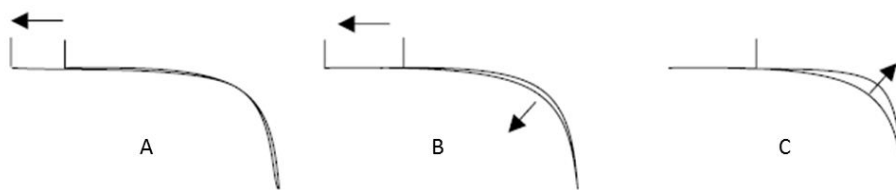
Figure 3.9: Alternatives for creating more *lead-in*

A. Keep the maximum elongation point of the old blend and start the new blend earlier
B. Decrease the elongation of the old blend and start the new blend earlier
C. Keep the end points of the old blend and extend the elongation

Finding the lead-in faces some special difficulties when seeking blendings between convex and concave curves. In these cases, the curves to be blended should already carry the information that they tend to change direction, i.e. if we extend the curves by rendering their functions with parameters greater than 1, they should become s-shaped. This character must already be carried within the evolution of the curve's curvature and its derivatives. Another particularity is that one can give a negative lead-in to make a corner look crisper.
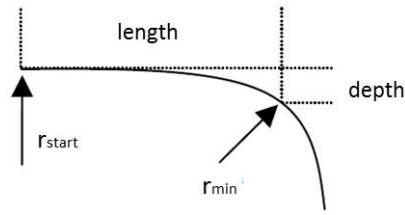
Starting with a true radius blending a lead-in could be characterized by two parameters (see Figure 3.10): How much of the main curve do we cut away until we reach the minimum radius (lead-in length), and how deep under the curve will we be then (lead-in depth). Those two parameters can be measured independently from each other, but the length is the more important one.

Also, the radius change (the difference of the radius at the start of the blending and the minimum blending radius) should be included, so that we yield a measure which is less dependent on the actual size of the model:

$$Lead\text{-}in = length/(r_{start} - r_{min}) \tag{3.11}$$

3.2.4.6   Sharpness/softness

The term is used to describe the properties of transitions between curves or surfaces. In styling, the term radius is generally used to indicate a somehow more rounded transition (a blending) between two curves or also surfaces. In general, (blending with) a small radius can be called sharp, and (blending with) a big radius can be called soft. Giving absolute values for sharpness or softness is in most cases less important than giving the difference of two curves. Then, making a sharper (softer) radius means to decrease (increase) the radius of the blend. It can also mean to create a blending with only G1-continuity or even G0-continuity instead of a soft G2 connection. The classical meaning of the G1 and G2 continuity refers to the tangency (first derivation) continuity and derivation of the tangency (second derivation) continuity, respectively. The G0 continuity refers to not having a gap between blending surfaces. The meaning of "big" and "small" depends on the sizes and proportions of the curves to be connected. This has to be taken into account, especially while working on scale models. Almost any property must be exaggerated there in order to achieve the same effects as in the full size model. The transition between two curves / surfaces can look hard, if there is not enough lead-in between curve and radius. When we give measures, we should concentrate on the minimum radius of a given blending curve and say that a sharp radius is a small radius, while a soft radius is a big one. The measure for softness considers the ratio of the minimum radius and the length (*L*) of the measured curve:

$$Softness = radius_{min} / L = 1 / sharpness \tag{3.12}$$

3.2.5 Synthesis

This list with terms for model features and their properties is used in styling work, and each application area puts emphasis on different concepts. Although styling is a very creative field of work, few terms are sufficient to communicate design intentions. There is something like a common language. The language is not unique, but it allows describing the changes of the model. The terms found can formally be described and measured.

3.3 Aesthetic properties of surfaces

Aesthetic shapes are usually designed by means of free-form surfaces. As explained, the main geometric quantities used by the designers to reach aesthetic surfaces are 2D and 3D curves. Although the curves give the main character of the shape, the overall impression of the shape in terms of aesthetic appearance is represented by the entire surface. Therefore, it is very important to also find a mapping of aesthetic properties directly to the surfaces, instead of only mapping them first on curves, and then generalizing them to the surfaces. In the previous section, the list of aesthetic properties identified in the FIORES-II project has been presented. This list of aesthetic properties together with their mathematical quantifications refers almost exclusively to curves and not to surfaces. For a given aesthetic property, the computed measure is appropriate for application to curves, whereas their application to surfaces is not that straightforward. Since the surfaces, as a geometric entity of a higher dimension compared to the curves, have more complex shape characteristics, the measure to quantify a given property would be much more complicated. The list of aesthetic properties proposed by FIORES-II project can be considered as a very good starting point in defining aesthetic properties of surfaces. Some of the aesthetic properties for curves (straightness, convexity/concavity and acceleration) can be extended to surfaces, whereas the others (tension, crown, and lead-in) are meaningless or vague in cases of their application to surfaces. Regarding the mathematical quantification of the property, their direct extension to surfaces is not possible. This is not possible simply because there are curve quantities that do not exist or do not have the same meaning for surfaces. Additionally, the mathematical quantification of a given curve property contains geometric quantities that do not exist or do not have the same meaning for surfaces. For instance, the straightness of a curve is defined by four curves quantities (eq. 3.NS = *non-straightness* = $\frac{C \cdot A \cdot L}{l^2}$ (3.2)). One of these quantities is the curve length (*l*) and it cannot be directly extended (uniquely) to surfaces. In other words, the same aesthetic property for curves and surface might not share (not be affected by) the same geometric quantities. Therefore, it is necessary to determine another way for their quantification. In order to be able to define a mathematical equation, it is required to know which geometric quantities of the surface are strongly related to a given property.

The objective of this thesis is to propose a framework to map aesthetic properties to surface shapes. The framework has been implemented and verified for the mapping of the extension of the straightness property of curves to surfaces. For surfaces, we do not speak of straightness, but the notion of flatness is more adapted.

3.4 Conclusion

The first section of this chapter has introduced the aesthetic-oriented free-form shape declarative design. It has discussed the possibility for integrating the aesthetic aspect of the product shape in the industrial design process, as well as some concepts relative to declarative modeling. The aim of the integration is to open new perspectives for developing high-level modification of geometric models. This first section has been organized in three parts. The first part introduced the declarative design concept that is considered to be an opposite modeling approach to the procedural modeling. The main difference is that the declarative modeling, while defining the object, uses more abstract notions than low-level geometric quantities. The higher-level of shape abstraction places the declarative modeling on the conceptual and semantic level. Daniel et al. (Daniel & Lucas, 1997) suggest that declarative modelers must have at their disposal, tools for describing, generating and understanding. The second part presented the target-driven design concept that aims at direct shape generation, satisfying a certain property (target). The declarative approach is considered as application-oriented and target-driven design approach. The third part is the aesthetic-oriented manipulation of 3D free-form shapes. This part introduces the aesthetic-oriented design as target-driven design, where the target is the aesthetic property. There are two general approaches to associating aesthetic properties to free-form shapes: mapping of aesthetic properties and definition of Aesthetic Curves and Surfaces. The latter approach is curvature based, less intuitive, and inconvenient in application for designing more complex shapes. Therefore, the approach of mapping of aesthetic properties has been adopted. The mapping of aesthetic properties can be performed by conducting interviews requesting non-professionals to evaluate a given set of shapes, with respects to a certain aesthetic property.

In the second section, the aesthetic properties of curves, proposed by the FIORES II Project have been presented. The list of aesthetic properties can be considered as a very good starting point in defining aesthetic properties of surfaces. Many of these properties can be directly extended to surfaces, whereas others cannot be extended directly. Although some of them can be extended to surfaces, none of them can use the same mathematical quantification. The mathematical quantification of the curve straightness cannot be directly extended to surface, as well as the geometric quantities related to the straightness. The investigation of which surfaces quantities are relevant, with respect to the flatness, is the first objective of this thesis.

The straightness of a curve is computed by using the measures of four geometric quantities (length, area, curvature and cord distance). The surfaces are much more complex geometric entities which can be described by much more geometric quantities than curves. For instance, there is no single value for the curvature at each surface point, but the principal curvature can be computed. This is later used to compute the Gaussian, Mean and Absolut curvature. The area below the curve, in 3D space for surfaces, is represented by the volume below the surfaces. Here, the volume below the surface can be computed with respect to many referential planes such as: the 3 origin planes (XY, XZ, YZ), and the planes of the minimal bounding box. Furthermore, the computation of the surface area requires appropriate triangulation. The projection of the surface can be done onto the three origin planes (XY, XZ, YZ) and the planes of the minimal bounding box. Moreover, the investigation for finding the relationships between the surface quantities and a given aesthetic property (e.g. flatness) is very difficult. Additionally, the identification of which surface quantities are relevant with respect to the flatness, is almost impossible without using more sophisticated techniques.

Therefore, in this thesis, we have decided to apply Artificial Intelligence (i.e. Machine Learning Techniques) in order to be able to determine which surface quantities are relevant with respect to the flatness. The Machine Learning Technique is more broadly presented in the next chapter (Chapter 4)

<div align="right">**Chapter 4**</div>

<div align="center">**Machine Learning Techniques**</div>

This chapter presents the bibliographic works related to the basics of Data Mining methodology (DM) and Machine Learning Techniques (MLTs). The chapter begins by introducing the Data Mining as a particular task in the entire Knowledge Discovering from Data (KDD) process (Section 4.1). Data Mining is a process designed for discovering hidden knowledge and structural patterns in a huge dataset by using wide range of methods and techniques. The most common techniques used for Data Mining are the Machine Learning Techniques. Furthermore, in Section 4.2, the categories of learning problems (e.g. supervised and unsupervised) have been presented. The next section (4.3) focuses on the use of WEKA workbench to identify classification rules and meaningful attributes. Depending on how the dataset of instances is labeled, the techniques for solving different classification problems (Section 4.4) can be divided in different groups for solving: 1. single-label classification (Section 4.4.1), 2. multi-label classification (Section 4.4.2), and 3. multi-dimensional classification (Section 4.4.3). Further in this chapter, (Section 4.5), the applications of MLTs for solving problems in various domains, including the mapping of aesthetics to 3D free-form shapes are presented. The last section (4.6) gives a synthesis of this chapter.

4   Machine Learning Techniques

4.1   Data Mining and Knowledge Discovery

Today, it is evident that the data flow increases rapidly every day for almost all domains related to the data processing and computer science. Therefore, the need for finding a way to mine and extract knowledge of the data is becoming even more crucial. The activity of mining in huge data is named Data Mining and its purpose is to solve problems by analyzing data already present in the databases. Dunham (Dunham H., 2003) defines the Data Mining (DM) as automatic (or semi-automatic) process for discovering structural patterns in a huge dataset using wide range of methods divided in five groups: machine learning, statistics, information retrieval, algorithms, and database (Figure 4.1). The structural pattern discovered must be understandable and meaningful in terms of revealing new knowledge.
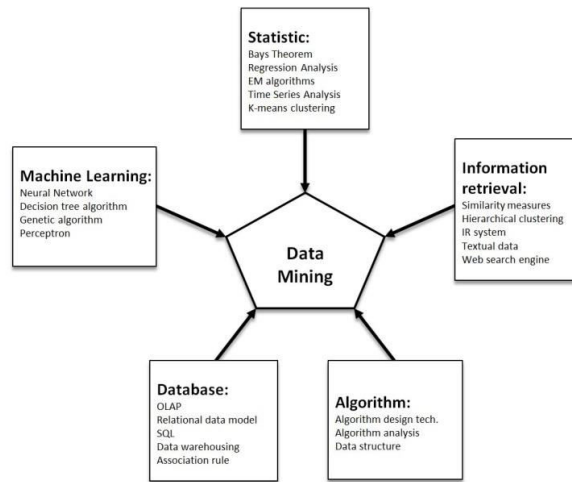
Figure 4.1: Classification of Data Mining methods (Dunham H., 2003)

Fayyad considers Data Mining (DM) as one of the phases of the KDD (Knowledge Discovering from Data) (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). In fact, the KDD consists of many activities carried out, aiming at discovering new and useful knowledge from a dataset where DM is one particular task in the entire KDD process (Figure 4.2). The DM step mainly concerns the means by which the patterns are extracted and collected from the data, using specific algorithms.
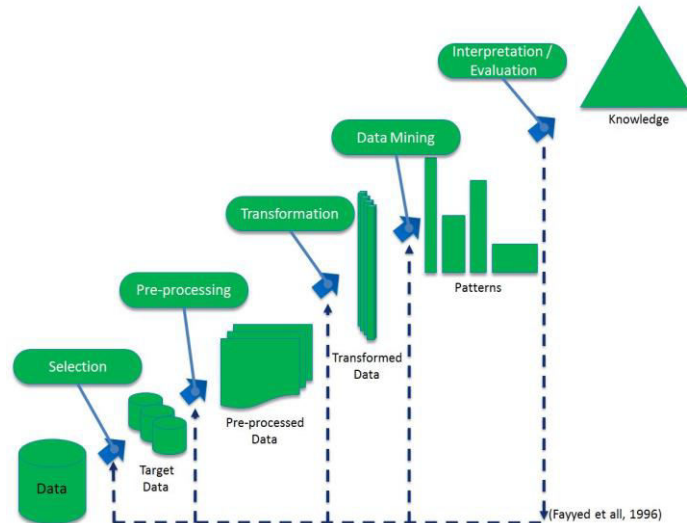
Figure 4.2: The Knowledge Discovery from Data – KDD (Fayyad, Piatetsky-Shapiro, & Smyth, 1996)

The KDD overall process depicted on Figure 4.2 helps understanding which additional steps have to be foreseen before applying any learning techniques. The KDD consists of five steps for extracting the knowledge: 1. data selection, 2. data pre-processing, 3. data transformation, 4. data mining and 5. interpretation of the results. There are many essential steps

to ensure that a useful knowledge is derived from the data (Fayyad, Piatetsky-Shapiro, & Smyth, 1996) :

1. **Selection** consists in creating of a Target Data. During this activity, the initial Data is being filtered by omitting some variables and focusing on a subset of variables (i.e. data samples), which will be later used for the DM process.
2. **Pre-processing** refers to Target Data preparation and structuring in order to obtain consistent data.
3. **Transformation** refers to the transformation of the Pre-processed Data using wide range of methods for data transformation and dimensionality reduction.
4. **Data Mining** consists in investigating patterns that are subject of interest. The patterns discovered are represented in appropriate form depending on the Data Mining objectives
5. **Results interpretation/evaluation** refers to the patterns interpreting and evaluating in order to reveal new and useful knowledge.

A direct application of data mining methods, i.e. without considering the above-mentioned steps, would be hazardous and would lead to a meaningless discovery characterized by invalid patterns and rules.

Of course, those steps strongly depend on the application domain (to be considered). It is therefore crucial to understand it, as well as to identify the relevant prior knowledge and the goal of the KDD process, from a customer's viewpoint. To be efficient, DM requires a complete understanding of the application, of the data, and mining methods. Actually, DM problems and tasks can be classified as follows (Michael, 2002):

1. **Diagnosis-** to define malfunctions and then propose solution.
2. **Pattern recognition-** to identify objects and shapes, colors, surfaces, texture, temperature, or any other attributes, using automatic means.
3. **Prediction-** to predict the behavior of an object
4. **Classification-** to train a classifier and assign a class to an unknown object.
5. **Clustering-** to split a heterogeneous set of objects into homogeneous subsets of objects that share common properties.
6. **Optimization-** to improve the performance of a function until finding an optimal solution.
7. **Control-** to manage the performance of a function to accomplish specified requests.

To solve those problems, DM can make use of Machine Learning Techniques (MLTs) that can be categorized by their usefulness and efficiency (Figure 4.3):

1. **Artificial Neural Networks** are widely used for DM problems in many disciplines, such as pathology, biology, statistics, image processing, pattern recognition, optimization of numerical analysis, as well as controlling systems.

2. **Genetic Algorithms**, such as evolutionary computation techniques, are well-known solving approaches for DM problems in chemistry, biotechnology, movement prediction, bio-informatics and adaptive control for working systems.

3. **Inductive Logic Programming** shows a restricted area of applications when compared to other MLTs. However, it has been applied to diagnosis (diseases diagnostic), classification and clustering problems, controlling robotics systems, etc.

4. **Rule Induction** shows some applicability for optimization (a good example would be semantic query optimization).

5. **Decision Trees** is a powerful DM tool to solve problems in most of real world cases (prediction, classification, etc.). Moreover, by using decision tree induction process, control rules can be derived.

6. **Instance-based Learning Algorithms** are defined as the generalizing of a new instance to be classified from the stored training examples, which is widely used for classification problems.
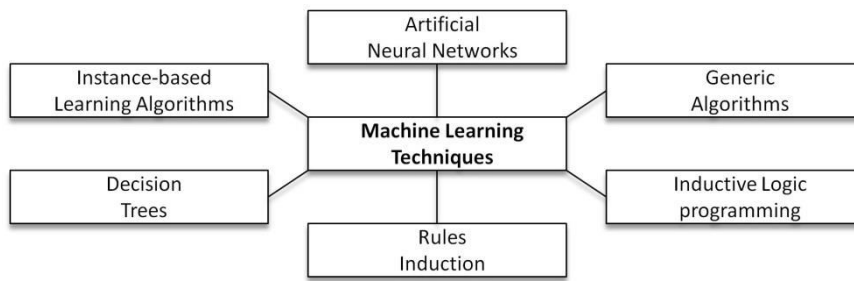
Figure 4.3: Machine Learning Techniques applied to Data Mining problems (Bramer, 1999)

Machine Learning and Data Mining often use the same methods and overlap (Figure 4.4). As explained, Data Mining consists in applying a variety of methods to extract patterns and knowledge from data. Arthur Samuel, founder and pioneer in Artificial Intelligence and Machine Learning, defines Machine Learning as a "field of study that gives computers the ability to learn without being explicitly programmed" (Simon, 2013).
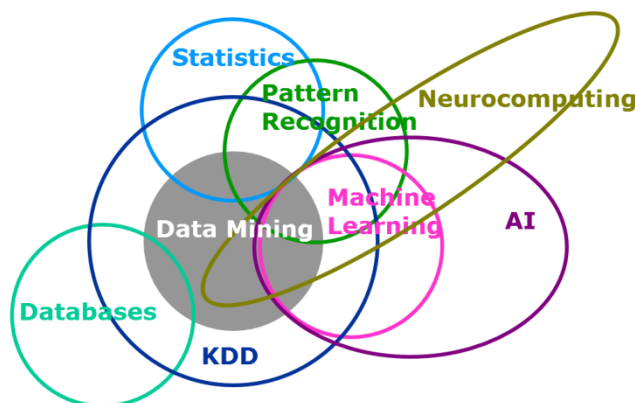
Figure 4.4: Illustration of the overlapping of different fields of studies (Mitchell-Guthrie, 2014)

60

Thus, the Machine Learning focuses on the prediction task, where the knowledge learned from past experiences (the training data) replicates to the new unknown instances, while Data Mining focuses on discovering unknown knowledge.

As introduced in Chapter 3, aesthetic properties of 3D shapes rely on numerous and sometimes complex geometric quantities. Even if the users can classify aesthetic shapes with respect to those aesthetic properties, they are unable to identify the relationships and rules that link those classes to the low-level geometric quantities. Thus, the use of automatic Machine Learning Techniques seems a good way to try to discover these links and relations between aesthetic properties and the geometric quantities, and it is the basic idea/hypothesis of this thesis.

4.2   Categories of Machine Learning Techniques

Machine Learning Techniques are considered as a core subarea of Artificial Intelligence (AI) that makes use of computer algorithms for learning from data in order to deal with some tasks. Artificial Intelligence (AI) has wide application in solving Data Mining (DM) problems. In general, tasks that address Data Mining problems can be also divided as follows (Figure 4.5):

1. **Predictive** tasks refer to Classification, Regression, and Prediction.
2. **Descriptive** tasks refer to Attribute Selection, Clustering, Association Rules, and Summarization.
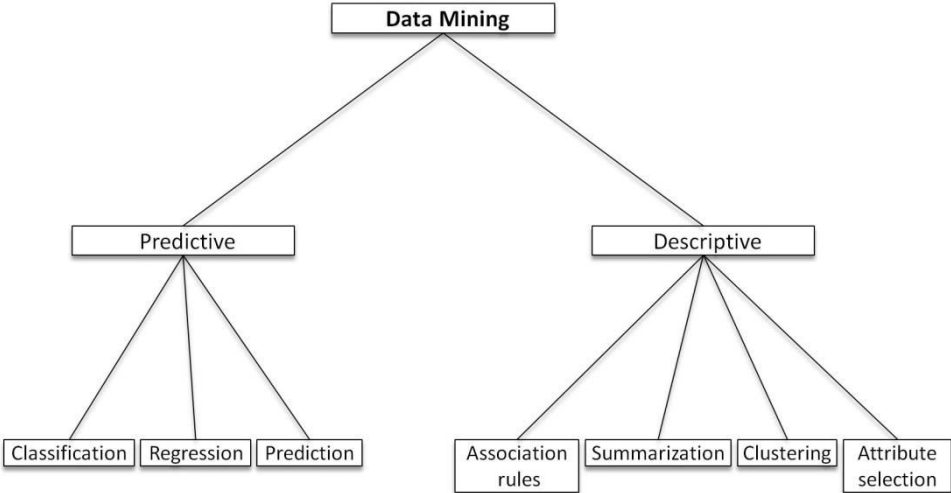
Figure 4.5: Groups of Data Mining problems into (Witten, Frank, & Hall, 2011)

Machine Learning Techniques provide very efficient algorithms to deal with all kinds of DM tasks. In relation to the learning task, there are two types of learning problems:

- **Supervised learning** that deals with predictive DM tasks where all instances are labeled by someone, i.e. evaluated and classified by him/her. Supervised learning is the task of using algorithms that allow computers to learn associations between instances and class labels. Supervision comes in the form of previously-labelled instances, from which an algorithm builds a model to automatically predict the labels for new instances. Previously-labelled instances are readily available in real world scenarios, usually in the form of human-annotation by an expert.

- **Unsupervised learning-** when the instances are not labeled, i.e. are not supervised, and when the objective is to find intrinsic relationships between instances. The unsupervised learning is much harder than supervised learning because it is much harder to make the computer to learn to perform some task without telling how to do it. The goal of the unsupervised learning is to find similarities and common properties between instances and group them in clusters. In fact, the clusters discovered can match with an intuitive classification.

4.3   Use of WEKA to identify classification rules and meaningful attributes

Since our objective is not to define new Machine Learning Techniques, it has been decided to adopt WEKA as the environment where to test the different algorithms on our pre-processed data, i.e. data referring to aesthetic curves and surfaces. The WEKA (Waikato Environment for Knowledge Analysis – University of Waikato, New Zealand) environment gather together a collection of the most used machine learning algorithms and data pre-processing tools designed in a way that provides quite flexible manipulations. (Witten, Frank, & Hall, 2011) has provided a comprehensive and detailed description of WEKA workbench. In addition, WEKA performances and particular functions are described in various papers such as (Aher & Lobo, 2011), (Garner, 1999), thus illustrating its use in different applications.

Actually, there exist three ways of using WEKA. The first one is to apply a learning method to a dataset and analyze its output in order to learn more about the data. The second one is to use a learned model in order to generate prediction of new instances, and the third one is to apply several different learners and compare their performance in order to choose which of them is the most suitable for prediction.

In addition, WEKA provides several interfaces: Explorer, Experimenter, Knowledge Flow and Simple CLI. For our purposes, the most useful interface is the Explorer one that reads a dataset from a so-called ARFF file and builds a learning model. The second WEKA interface is the Experimenter one that is designed to help to answer basic practical questions

when applying classification or regression techniques: which methods and parameters' values are to be used for the given problem? In addition, the Experimenter interface provides a good environment to compare a variety of learning techniques. The Knowledge Flow interface allows the manipulation with boxes that represent learning algorithms and data sources around the screen, in order to design the desired configuration. In other words, it helps the design of the structure of the data flow with simple connection of boxes that represent data source, appropriate tools, learning algorithms, evaluation methods and visualization modules. Simple CLI provides direct access to all learning algorithms through a command-line interface (CLI). By selecting the Simple CLI from the WEKA interface, it brings up a plain textual panel with a line at the bottom on which we can enter commands.

In our case, the Explorer interface is used to read and prepare the data, to apply the learning algorithms and to visualize the results. Using the WEKA environment, one should be able to:

- Identify most meaningful attributes characterizing our Data Mining problem, i.e. which geometric quantities best characterize aesthetic properties of curves and surfaces;
- Create and customize a classification model using Machine Learning algorithms and associated control parameters.

The next section introduces the different algorithms that can be used to create and customize our classification model.

4.4   Different classification techniques

The WEKA Explorer interface gathers together a collection of the most-used MLT organized in 6 groups: preprocess, classify, cluster, associate, attribute selection, and visualization. Furthermore, the pre-process and visualization groups contain filtering and discretization algorithms as well as windows to visualize the learned model (e.g. trees). The other four groups respectively, address the supervised learning (classify group) to solve predictive data mining problems, unsupervised learning (cluster group) for descriptive data mining problems, so as the associate and attribute selection groups. Of course, it is possible to access and modify all the control parameters of those techniques. Table 1 lists the available WEKA classifiers, as well as a brief description of each of them. The classifiers can be categorized according to 7 categories: Bayesian, function, lazy, multi-instance, trees, rules, and a final miscellaneous category.

Classifiers	Name	Description
Bayes	BayesNet	Learns Bayesian nets
	ComplementNaiveBayes	Builds a Complement Naïve Bayes classifier
	DMNBText	Discriminative multinomial Naïve Bayes classifier
	NaiveBayes	***Standard probabilistic Naïve Bayes classifier***
	NaiveBayesMultinomial	Multinomial version of Naïve Bayes
	NaiveBayesMultinomial-Updateable	Incremental multinomial Naïve Bayes classifier
	NaiveBayesSimple	Simple implementation of Naïve Bayes
	NaiveBayesUpdateable	Incremental Naïve Bayes classifier
Functions	LibLINEAR	Wrapper classifier for using the third-party LIBLINEAR library for regression
	LibSVM	Wrapper classifier for using the third-party LIBSVM library for support vector machines
	Logistic	Builds linear logistic regression models
	MultilayerPerceptron	***Backpropagation neural network – NN***
	RBFNetwork	Implements a radial basis function network
	SimpleLogistic	Builds linear logistic regression models
	SMO	***Sequential minimal optimization algorithm for support vector classification – SVM***
Lazy	IB1	Basic nearest-neighbor instance-based learner
	IBk	***k-nearest-neighbors classifier k – NN***
	KStar	Lazy Bayesian Rules classifier
	LWL	General algorithm for locally weighted learning
MI	CitationKNN	Citation KNN distance-based method
	MISMO	SMO using multi-instance kernels
	MIWrapper	Applies single-instance learner using the aggregating-the-output approach
	SimpleMI	Applies single-instance learner using the aggregating-the-input approach
Misc	HyperPipes	Extremely simple, fast learner based on hypervolumes in instance space
	VFI	Voting feature intervals method, simple and fast
Rules	ConjunctiveRule	Simple conjunctive rule learner
	DecisionTable	Builds a simple decision table majority classifier
	DTNB	Hybrid classifier combining decision tables and Naïve Bayes
	JRip	***Ripper algorithm for fast rule induction – DR***
	Nnge	Nearest-neighbor method of generating rules
	OneR	1R classifier
	PART	Obtains rules from partial decision trees built using J4.8
	Ridor	Ripple-down rule learner
	ZeroR	Predicts the majority class or the average value
Trees	BFTree	Builds a decision tree using a best-first search
	DecisionStump	Builds one-level decision trees
	FT	Builds a functional tree with oblique splits and linear functions at the eaves
	J48	***C4.5 decision tree learner (C4.5 revision 8)***

J48graft	C4.5 with grafting
LADTree	Builds multiclass alternating decision trees
LMT	Builds logistic model trees
NBTree	Builds a decision tree with Naïve Bayes classifiers
RandomForest	Constructs random forests
RandomTree	Constructs a tree that considers a given number of random features
REPTree	Fast tree learner that uses reduced-error pruning
SimpleCart	Decision tree learner using CART's minimal cost
UserClassifier	Allows users to build their own decision tree

This table of learning algorithms is just a small part of the entire set of learning algorithms that have been used in Artificial Intelligence (AI). Of course, this work does not aim at verifying or testing all existing learning algorithms, but rather it aims at investigating which of them are most suitable for our application, i.e. to discover interesting relationships between aesthetic properties and geometric quantities.

Many authors have been testing a large variety of learning algorithms in different application fields and several ranking of the 'best' algorithms have been suggested. (Wu, et al., 2008) presents the 10 best learning algorithms: **C4.5 (J48)**, **SVM**, k-Means, Apriori, EM, **kNN**, **Naïve Bayes**, PageRank, AdaBoost, and CART. In this work, the first step of the identification process was the invitation to the IEEE ICDM Research Contribution Award and ACM KDD Innovation Award. During the survey, each participant has been asked to nominate up to 10 well-known learning algorithms in Data Mining. Further on, the participants have been requested to provide the following information: (a) to name the algorithms, (b) to give a short justification of the suggested ranking, and (c) to present a reference for each of the proposed learning algorithms. By removing the nominated algorithms with less than 50 citations, a list of 18 nominees was designed. Following, this list of nominees was given to the Program Committee of KDD-06, ICD'06, ADM'06, and ACM KDD to vote up to 10 well-known algorithms from the 18 algorithms candidate list. (Kotthoff, Gent, & Miguel, 2010) presents a comprehensive comparison of machine learning algorithms and techniques tacking algorithm selection. It evaluates the performance of a large number of different techniques on data sets used in the literature. Most of the learning algorithms used in the comparison are algorithms available in WEKA, and the top of the well-performing contains LADTree, LinearRegrassion, **IBk(kNN)**, **J48**, **JRip**, RandomForest, **SMO(SVM)**, LibSVM, GaussiaProcesses. Furthermore, (Caruana & Niculescu-Mizil, An Empirical Comparison of Supervised Learning Algorithms, 2006) and (Shhab, Guo, & Neagu, 2001) propose a comparison of few supervised learning algorithms tested over a set of different learning problems. (Caruana & Niculescu-Mizil, An Empirical Comparison of Supervised Learning Algorithms, 2006) presents a number of supervised learning algorithms that have been widely used in the last decade, and also presents an empirical comparison between the following 10 supervised learning algorithms: **SVM**, Neural Nets (**NN**), Logistic Regression, Naïve Bayes (**NB**), Instante-based learning (**kNN**), Random forests, decision trees (**DT**), bugged trees, boosted trees, and boosted

stumps. (Shhab, Guo, & Neagu, 2001) intends to propose a study on AI techniques widely used to solve specific DM problems, as suggested by research papers published in the last 10 years. They also give experimental results of their four most efficient supervised learning algorithms: Instance-based Learner (**kNN**), Decision Trees (**J48**), Rule Introduction (**JRip**) and Artificial Neural Networks (**ANN**) tested over ten dataset for toxicity prediction. (11Ants Analytics, 2014) is a research spinoff of the Waikato University (founders of WEKA) that has commercialized a technology to automate the production of predictive models. The result combines a library of 11 machine learning algorithms that are the most widely used WEKAs algorithms: Decision Tree (**J48**), Naïve Bayes (**NB**), Nearest Neighbors (**kNN**), Random Forest, Gaussian Processes, Logistic Regression, Model Tree, Ridge Regression, Logit Boost, Support Vector Machine (**SVM**), PLS.

The previously listed machine learning algorithms are the most widely used to solve single-labeled learning problems (supervised learning of a data that are labeled with one label). In case of supervised learning problems, i.e. when the classification is performed through interviews, each instance of the database is associated with more than one label (multiple labeled instances). This is the case in our application. Therefore, it is essential to know which learning algorithms are also convenient for solving multi-label classification problems. (Read, 2010) gives a very exhaustive analysis of solving multi-labeled classification problems, and introduces the problem transformation methods for transformation of multi-label classification into single-label classification. This will be further detailed in Section 4.4.2. Table 2 shows a list of well-known single-label classifiers that have been used in the multi-labeled literature; either employed by problem transformation methods, or in modified forms as algorithm adaptation methods.

Table 2: A list of well-known single-label classifiers (Read, PhD thesis: Scalable Multi-label Classification, 2010)

Key	Name [citation]	(WEKA implementation)
NB	Naïve Bayes (John and Langley, 1995)	NaiveBayes
SVM	Support Vector Machine (Platt, 1999)	SMO
DT	Decision Tree (Quinlan, 1986)	J48
kNN	k – Nearest Neighbor (Wang et al., 2000)	IBk
NN	Neural Network (Haykin, 1998)	MultilabelPerceptron
DR	Decision Rules (Cohen, 1995)	JRip

After analyzing the learning algorithms that have been identified by many other authors, it is not difficult to summarize and conclude that the algorithms **Decision Tree (J48)**, **k – Nearest Neighbors (kNN)**, **Support Vector Machine (SVM)**, **Bayes Naïve (BN)**, **Decision Rules (JRip)** fit both the single-label and multi-label classification problems. Since, there is no explicit indication of which among these learning algorithm is the most convenient in our application, they all have been considered and will be further detailed in the following sections.

4.4.1   Single label classification

4.4.1.1   C4.5 decision trees or J48

The C4.5 algorithm of (Quinlan, 1993) works like a classification tree. It is called J48 in WEKA and is derived from the divide-and-conquer algorithm. The divide-and-conquer algorithm is based on recursion and divides the initial problem into many sub-problems that are similar to the initial problem. Such created sub-problems are solved (conquer) recursively and the solutions to these sub-problems are combined to solve the initial problem. Similarly to the divide-and-conquer (divide-conquer-combine) algorithm, C4.5 algorithm is applied to construct a decision tree. The tree generation starts by selecting an attribute to be the root node, and makes one branch for each possible value of the attribute. The division of the entire set of instances into subsets is done for each value of the attribute. This process is repeating recursively for every branch until all instances at a node have only one class, and the development of the tree stops. The open question for starting the development of the tree is how to determine which attribute to be used at the root node. When selecting the root node, attention has to be paid on the fact that the tree has to be as small as possible. Therefore, C4.5 uses the information gain that minimizes the total entropy (information theory) of the subsets. Attributes can be either numeric or nominal, and this determines the format of the test outcomes. This algorithm works best in cases where the attributes are numerical, which is the case in our application.

4.4.1.2   IBk or k – Nearest Neighbors (k-NN) classification

The k – Nearest Neighbors (k-NN) algorithm is a predictive lazy learning method. It is called IBk in WEKA. When a new instance is presented to the model, the algorithm predicts the class, using the majority class of the most similar training instances of k stored in the model. An obvious drawback of this approach is that many test records will not be classified because they do not match exactly any of the training records. Thus, (Tan, Steinbach, & Kumar, 2006) have designed a more sophisticated classification approach called k-NN (k - Nearest Neighbor) that finds a group of k instances in the training set that are closest to the test instances. The k-NN bases the assignment of the class of the instance by using a simple majority vote of the k nearest neighbors. The k-NN approach consists of three key elements: a set of classified instances, the computation of the distances between instances, and the number of the nearest neighbors – k. in order to assign a class to an unclassified instance, the distance between this instance and the classified instances is computed, then the k-nearest neighbors of the unclassified instance are identified, and the classes of the k – nearest neighbors are used to determinate the class of the unclassified instance. The perfor-

mance of the k-NN is highly affected by the choice of the number of the nearest neighbors. On the one hand, if the value for k is too small, it makes the classification of unclassified instance very sensitive to noise instances. On the other hand, if the value for k is too large, then the neighborhood may consist of too many instances (even entire set of instances) from other classes. The k-NN classification is easy to understand and to implement. It is particularly well suited for multi-modal classes, as well as application in which an instance can have many class labels.

4.4.1.3 SMO or Support Vector Machine (SVM)

Support Vector Machine (SVM) introduced by (Vladimir, 1995) is considered a must-try classification method because it represents a very accurate and robust method between all well-known algorithms. It is called SMO in WEKA. This method is described as a very simple linear model that can be used for solving classification problems where all attributes are numerical. The main shortcoming is that the linear modeling can only represent linear boundaries between the classes, which consider them as great simplification for many applications where linear bordering between classes does not exist. In case of a two-class classification problem, the goal of SVM is to approximate the 'best' classification function to differentiate between the instances of the two classes in the training set of instances. The geometric interpretation of the 'best' classification function, in linearly separable set of instances, corresponds to a separating line (hyperplane) that passes through the middle of the two classes and divides the two. After this classification function is defined, the new unknown instance can be classified only by testing on which side of the function the instance is. The geometrical definition of the classification function helps us to find a solution in case where there are infinite numbers of hyperplanes. A more recent strategy to address this problem of learning an SVM is to consider it as a problem of finding an approximate minimum enclosing ball of a set of instances. This learning algorithm is very suitable in application where complex correlation between the attributes exists, resulting in creation of a very efficient classifier. The better performance of the algorithm is in case where the attributes are nominal, but it is not less accurate when training classifier use numerical attributes. One of the initial drawbacks of SVM is its computational inefficiency. However, this problem can be solved with great success.

4.4.1.4 NaiveBayes or Naïve Bayes (NB)

The Naïve-Bayes (NaiveBayes in WEKA) classifier is based on the Bayesian theorem introduced by (George & Langley, 1995). The classification method based on the Bayesian theorem investigates the relations between attributes and the class of each instance so as to

deduce a conditional probability. The conditional probability of each class is derived by counting how many times one class appears with respect to the total number of instances in the training dataset (i.e. a prior probability). The initial assumption in this classification method is the independence of the attributes to each other which makes it a very simplistic one. Under this assumption, the conditional probability is computed by multiplying the probabilities of each separate attribute which makes this classification a fast and very simple process. Naïve Bayes has achieved good results in many cases, even when this assumption is violated. This method is important because it is very easy to construct, and there is no need of additional set up parameters for specifying the classification schemes. It is easy to interpret, so users, unskilled in classifier technology, can understand how this classification occurs. General discussion of the Naive Bayes method and its merits are given in (Hand & Yu, 2001) and (Jamain & Hand, 2005). Finally, maybe the NaiveBayes is not the best classification method in any particular application, but it is flexible to use, robust and performs quite well. Since it is very easy to construct classifier and can be readily applied to huge datasets, we intend to test its capacity to create a classification model, as accurate as possible, for classifying 3D shapes.

4.4.1.5 RIPPER or Decision Rules (JRip)

With the commercial application of data mining methods, increased attention is given to decision trees (Quinlan, 1993) and decision rules (Cohen, Fast Effective Rule Induction, 1995), which are closely related to each other. These techniques may perform well and have potential to give insight to the interpretation of data mining results, for example in marketing efforts. The methods for induction of decision tree are much more efficient than those for induction of decision rules, even if the latter starts by inducting decision tree. After numerous modification of the decision tree (C4.5) based algorithms, the creation of the RIPPERk has been proposed (Cohen, Fast Effective Rule Induction, 1995), which is very competitive with C4.5 regarding the efficiency on a large set of instances. Algorithms that learn sets of rules have many desirable characteristics. The induction of set of rules is relatively understandable for people and they outperform decision trees on many problems. Rule sets have a natural and familiar first order version, but one weakness with rule learning algorithms is that they often scale relatively poorly with the sample size, particularly on noisy data (Cohen, Efficient pruning methods for separate-and-conquer rule learning systems, 1993). Given the prevalence of the large noisy dataset in real-world application, this problem is of critical importance. Since this learning algorithm has very similar performance to the classification trees C4.5, with respect to the classification error but much more accurate on large datasets, which is the case in our application, we decided to test its performance for classifying 3D free-form shapes.

4.4.1.6 Training a classification model (classifier)

Classification is a supervised learning problem, which maps a data item into predefined classes. There are various classification algorithms proposed in literature, but as explained earlier, in our work we are focusing on the 5 well-known single-label classification algorithms. Classification is a method where one can classify future data into known classes. In general, this approach uses a training data to build a model, and a test set to evaluate it. The accuracy of supervised classifications is much better than the one of unsupervised classifications and it strongly depends on the prior knowledge, i.e. the training data. The efficiency of classification algorithms can be compared once the classification accuracies are known. The process flow for computing the classification accuracy is given in Figure 4.6. Before training a classifier, a testing set is required. Generally, classification algorithms offer two ways of providing the testing set: internal and external provisions. On one hand, the internal provision of the testing set consists in applying appropriate splitting algorithms so as to divide the initial dataset into two (train and test) sets. On the other hand, the external provision of the test set consists in supplying the classification algorithm with another additional dataset that is not a part of the initial dataset.
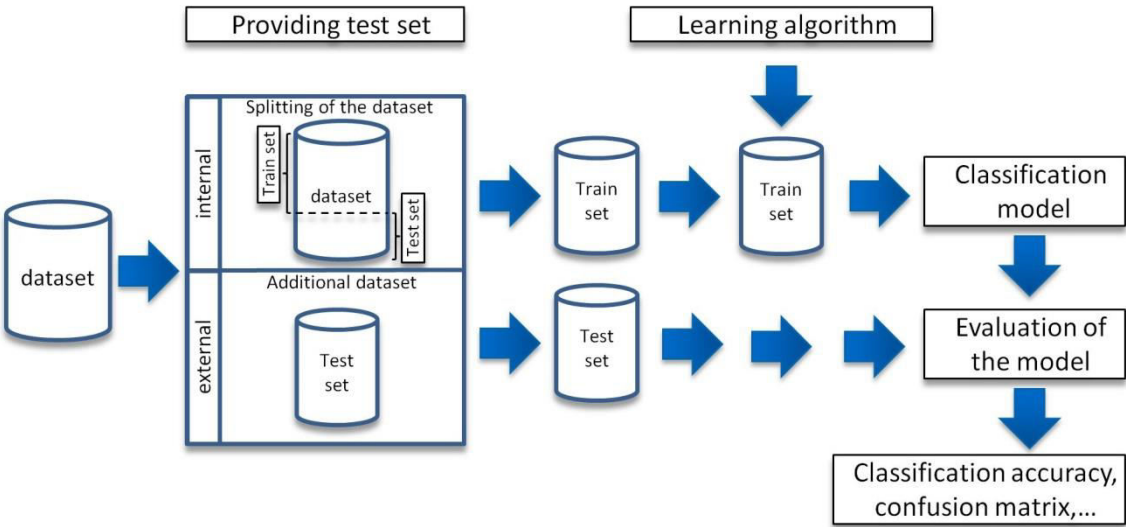
Figure 4.6: Evaluation of learning algorithms

The difference between those two manners lies in the fact that the testing set is either taking part in the initial dataset or not. In the second case, it is provided from somewhere else. The train set is being used to train a classification model, while the testing set is used to evaluate the classification model, resulting in computation of the classification accuracy, the confusion matrix and so on (Figure 4.6).

The internal provision of the testing set is carried out by applying a splitting algorithm that divides the initial dataset in two sets according to given ratios. Generally, two splitting strategies can be adopted: percentage split and cross-validation (figure 9). Regarding the division of the initial dataset, it is very important to decide on how to divide the initial dataset into training and testing set of instance (the splitting ratio). For the percentage split strategy, it is common to use one-third of instances for testing, and use the remaining two-thirds of instances for training. Representative training (or testing) sample means each class in the initial dataset should be represented in about the right proportion in the training and testing set. In general, it is not possible to be sure whether a sample is representative or not. For instance, if all instances with a certain class were omitted from the training set, then it is less clear that the classifier will not perform well on such instances of the dataset. Therefore, it has to be ensured that a random sampling is done in a way that guarantees that each class is properly represented in both the training and test sets, so-called stratification.

In addition, a more general way to reduce the influence caused by the chosen instances is to repeat the whole process, training and testing, several times, with randomly identified instance. At each iteration, a certain proportion (for instance, two-third) of the dataset is randomly selected for training, possibly with stratification, and the remaining is used for testing. In cross-validation method, a fixed number of folds, or partition, of the data is specified by the user. If the data is divided randomly into 10 parts (ten-fold), then the nine-tenth (9/10) of the instances are used to train the classifier, whereas the remaining one-tenth (1/10) is used for testing the classifier. Thus, the same procedure is repeated 10 times on different training sets and each time the classification accuracy is computed. To overall classification accuracy is computed by averaging the classification accuracy of all repetitions. The experiences of many classification tests have shown that 10 folds cross-validation gives the best estimation of error (Witten, Frank, & Hall, 2011). Thus, in our approach, the 10 folds cross-validation method will be used to evaluate the learning algorithms.

The external provision of the testing set is carried out by supplying the classification algorithm with an additional dataset. The additional testing set has to contain exactly the same attributes and classes as the training data, and it also has to represent the same process or phenomena as in the training data. This training set evaluation mode is also applicable in cases where we want to investigate whether there is a common judgment of a given event. For instance, a certain event is observed by different persons and they are requested to classify the event (e.g. perception of a given aesthetic property in our case). By training a classifier over the classification of one person and later testing its classification model with the classification of the remaining participants, it can be checked if this participant shares the same perception as the other participants.

4.4.1.7 Classification efficiency analysis

In order to introduce the various tools to analyze the efficiency of classifiers, a simple open source data set has been analyzed, using the C4.5 decision tree algorithm. The efficiency of the chosen learning algorithm (C4.5) will be tested according to the previously introduced training options:

1. **Use training set**: the efficiency of the classifier is evaluated using the same set of instance that was trained on;
2. **Supplied test set**: the efficiency of the classifier is evaluated using different set of instances loaded from another file ;
3. **Cross-validation**: the efficiency is evaluated using cross-validation with a fixed number of folds;
4. **Percentage split**: the efficiency is evaluated while analyzing how well the classifier predicts a certain percentage of the data which is held out for testing.

No matter which evaluation method is used, the output classification model is always the one built from all training data. In this section, where efficiency analysis tools are introduced, the classifiers have been trained to predict a single-labeled class. Once a classifier has been learned, several analyses are accessible and can be summarized as follows. Even if those information have been obtained using WEKA in the present case, the available analysis tools are quite generic and can be found in other software:

1. **Run information**: listing the main information related to the learning algorithm options, relation name, the number of instances, the attributes and test mode for evaluation of the classifier.

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:    iris
Instances:   150
Attributes:  5
             sepallength
             sepalwidth
             petallength
             petalwidth
             class
Test mode:10-fold cross-validation
```

The algorithm used – J48
The relation name – 'iris'
Number of instances – 150
Number of attribute – 5
The list of attributes (sepallength, sepalwidth, petallength, petalwidth) as well as the output class

Selected test mode: 10-fold cross-validation

Figure 4.7: Run information of the classifier

2. **Classifier model**: representing the classification model that was produced using the full training data.

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)

Number of Leaves  :     5

Size of the tree :      9


Time taken to build model: 0 seconds
```

Classification model represented in textual form that was trained on the entire training set of instances. The first split is on the 'petalwidth' attribute, whereas on the second level, the splits are on the 'petallength'.
In the tree structure, the class label that is assigned to a particular leaf is followed by the number of instances in the training set that reach the leaf.
Below the tree structure, there is a number of leaves (which is 5) and the number of nodes in the tree, i.e. the size of the tree (which is 9).

The program also gives the time needed to build the model, which is less than 0.05 seconds.

Figure 4.8: Classifier model - decision tree

3. **Summary**: Listing the main information related to the classification accuracy (correctly and incorrectly classified instances) and statistics, with respect to the chosen test mode.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96     %
Incorrectly Classified Instances         6                4     %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
Relative absolute error                  7.8705 %
Root relative squared error             33.6353 %
Total Number of Instances              150
```

This part lists the main information related to the predictive performance. The set of measures is derived from the training data.
In this case, 96% of 150 training instances have been classified correctly, while only 4% of the training instances have been classified incorrectly. In addition to classification error, there are the evaluation output measures derived from the class probability assigned by the tree such as: mean output error (0.035), the root mean squared error (0.1586), the relative error (7.8705%) and the root relative squared error (33.6353%).

Figure 4.9: Stratified cross-validation of the model

4. **Confusion matrix**: shows the distribution of how many instances have been assigned to each class. In this example, on the 50 instances initially classified Iris-setosa, 49 are well classified as Iris-setosa by the classifier, and only 1 is badly classified as Iris-versicolor.

```
=== Confusion Matrix ===

 a  b  c   <-- classified as
49  1  0 |  a = Iris-setosa
 0 47  3 |  b = Iris-versicolor
 0  2 48 |  c = Iris-virginica
```

Figure 4.10: Confusion Matrix

The confusion matrix is used to measure the performance of the classification problem. It distinguishes right diagonal TP (true positive) and TN (true negative) correctly

classified instances from FP (false positive), and FN (false negative) incorrectly classi-
fied instances (Gupta, Malviya, & Singh, 2012). The total number of instances is the
sum of the correctly and incorrectly classified instances (Figure 4.10).
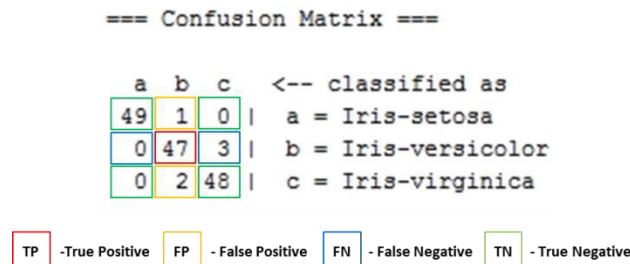
Figure 4.11: A structure of the Confusion Matrix

The classification efficiency analysis includes the computation of the following rates,
which helps us to calculate the parameter given in detailed accuracy by class (Figure 4.11)
such as Recall, Precision, F-Measure, and Accuracy:

True positive rate: TP rate $= \dfrac{TP}{TP+FN}$

False positive rate: FP rate $= \dfrac{FN}{TP+FN}$

True negative rate: TN rate $= \dfrac{TN}{FP+TN}$

False negative rate: FN rate $= \dfrac{FP}{FP+TN}$

5. **Detailed accuracy by class**: a more detailed per class breakdown of the classifiers' pre-
diction accuracy can also be obtained.

```
=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
                0.98      0         1           0.98     0.99        0.99       Iris-setosa
                0.94      0.03      0.94        0.94     0.94        0.952      Iris-versicolor
                0.96      0.03      0.941       0.96     0.95        0.961      Iris-virginica
Weighted Avg.   0.96      0.02      0.96        0.96     0.96        0.968
```

Figure 4.12: Detailed accuracy by class

Ratio of correctly classified instances to the total number of instances of a given class:

Recall: Recall $= \dfrac{TP}{TP+FN}$

Ratio of correctly classified instances to the entire number of instances classified with
same class

Precision: Precision $= \dfrac{TP}{TP+FP}$

74

Combination of Recall and Precision. It is also defined as harmonic mean of Precision and Recall.

F-Measure: $\qquad$ F-Measure $= \dfrac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$

ROC curve is plotting of FP Rate (on the x axis) against TP Rate (on the y axis) for the entire training set. The first point of the ROC curve is represented by computing the FP Rate and TP Rate of the first instance whereas the last point of the ROC curve represents the FP Rate and TP Rate of all instances in the dataset. By computing of all intermediate points for FP Rate and TP Rate, the ROC curve can be plotted. ROC area is the area under the ROC curve.

Ratio of correctly classified to total number of instances.

Accuracy: $\qquad$ Accuracy $= \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$

4.4.1.8 Relevant attribute selection

Generally, if a huge set of data is characterized by many attributes whose importance is not known, then Feature Selection (FS) can be applied. FS allows identifying which attributes can be omitted without affecting the accuracy of the learned model. The presence of less relevant and redundant features will affect the performance of the classification in terms of classification accuracy and time needed for creating the classification model. Attribute Selection aims at reducing the dimensionality of the patterns for solving classification problems by selecting the most informative instead of irrelevant and/or redundant features. In the case of a pattern recognition problem, the aim of FS is to find the smallest subset of features that are highly correlated with the learning model, so that it/so this improves its learning performance. Theoretically, this can be accomplished by finding all possible subsets of features and testing their correlation with the learning problem. This approach is known as exhaustive FS. The feature selection problem consists of two step activities: 1. evaluation function and 2. search methods. The evaluation function estimates the goodness of all subsets of features in making difference between classes, and is divided into two groups: filters and wrappers (Figure 4.13). Filters measure the relevance of the feature subsets independently of any classifier, and the undesirable features are filtered out of the data before the learning begins (Hall, Correlation-based Feature Selection for Machine Learning, 1999) (Akadi, Ouardighi, & Aboutajdine, 2008). The filter evaluation measures are faster than wrapper and can handle large datasets (Dash & Liu, 1997). The most popular filter-based measures are distance measures, consistency measures, and information measures. The wrappers use the learning algorithm along with a statistical resampling technique, such as cross-validation, to estimate the final accuracy of feature subsets (Kohavi, Wrappers for

Performance Enhancement and Oblivious Decision Graphs, PhD thesis, 1995). The wrapper approach requires a search space, operators, a search engine (hill-climbing and best-first), and an evaluation function (Kohavi & John, Wrappers for feature subset selection, 1997). The best-first search engine has been tested over wide range learning problems and has confirmed that they improve NaiveBayes and C4.5 in terms of accuracy and in comprehensibility, as measured by the used number of features.
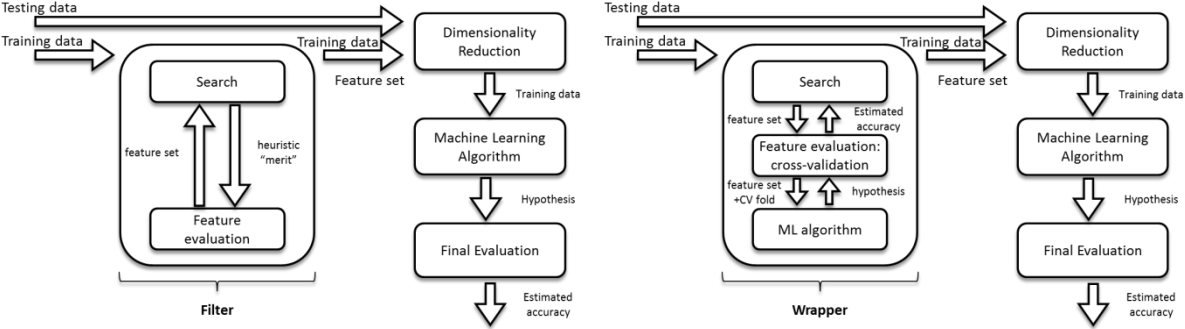

Figure 4.13: Filter and Wrapper feature selectors (Hall, Correlation-based Feature Selection for Machine Learning, 1999)

Attribute selection is normally performed by searching the space of attribute subsets, evaluating each one. This is achieved by combining one of the 6 attribute subset evaluators listed in Table 3 with one of the 10 search methods listed in Table 4. Another approach for feature selection is to evaluate the attributes individually and rank them omitting the attributes that are not correlated at all with the learning problem. This is achieved by selecting one of the 11 single-attribute evaluators listed in Table 3 and using the ranking method identified in Table 4. All the combinations are not appropriate.

Subset evaluators take a subset of attributes and return a numerical measure that guides the search. Witten et al. (Witten, Frank, & Hall, 2011) suggests that the ***CfsSubsetEval*** estimates the predictive ability of each attributes separately, favoring those sets of attributes that are highly correlated with the learning problem and with low correlation between each other. *ConsistencySubsetEval* evaluates the degree of consistency of the attribute sets when the training set of instances is projected onto the set. Whereas the previously mentioned subset evaluators are filter methods of attribute selection, *ClassifierSubsetEval* and *WrapperSubsetEval* are wrapper methods. Search methods traverse the attribute space to find a good subset. Quality is measured by the chosen attribute subset evaluator. ***BestFirst*** performs greedy hillclimbing with backtracking. You can specify how many consecutive non-improving nodes have to be encountered before the system backtracks. Subsets that have been evaluated are cached for efficiency.

InfoGainAttributeEval evaluates attributes by computing their information gain regarding the assigned class. Single-attribute evaluators are used with the **Ranker** search method to generate a ranked list from which Ranker discards a given number. They can also be used in the *RankSearch* method. For more detailed and exhaustive information about the attribute evaluation algorithms and search methods, the readers are invited to refer to (Witten, Frank, & Hall, 2011).

Table 3: Attribute evaluation methods for Attribute Selection

Evaluator	Name	Description
Attribute Subset Evaluator	***CfsSubsetEval***	Consider predictive value of each attribute individually.
	ClassifierSubsetEval	Use a classifier to evaluate the attribute set
	ConsistencySubsetEval	Project training set onto attribute set and measure consistency in class values
	CostSensitiveSubsetEval	Makes its base subset evaluator cost sensitive
	FilteredSubsetEval	Apply a subset evaluator to filtered data
	WrapperSubsetEval	Use a classifier plus cross-validation
Single-Attribute Evaluator	ChiSquaredAttributeEval	Compute the chi-squared statistic of each attribute with respect to the class
	CostSensitiveAttributeEval	Make its base attribute evaluator cost sensitive
	FilteredAttributeEval	Apply an attribute evaluator to filtered data
	GainRatioAttributeEval	Evaluate attribute based on gain ratio
	InfoGainAttributeEval	Evaluate attribute based on information gain
	LatentSemanticAnalysis	Perform a latent semantic analysis and transformation
	OneRAttributeEval	Use OneR's methodology to evaluate attributes
	PrincipalComponents	Perform principal components analysis and transformation
	ReliefFAttributeEval	Instance-based attribute evaluator
	SVMAttributeEval	Use a linear support vector machine to determine the value of attributes
	SymmetricalUncertAttributeEval	Evaluate attribute based on symmetrical uncertainty

Table 4: Search Methods for Attribute Selection

Search	Name	Function
Search Method	*BestFirst*	Greedy hill climbing with backtracking
	ExhaustiveSearch	Search exhaustively
	GeneticSearch	
	GreedyStepwise	Search using a simple genetic algorithm
	LinearForwardSelection	Greedy hill climbing without backtracking; optionally generate ranked list of attributes
		Extension of BestFirst that considers a restricted
	RaceSearch	number of the remaining attributes when expanding the current point in the search
	RandomSearch	Use race search methodology
	RankSearch	Search randomly
		Sort the attributes and rank promising subsets using an attribute subset evaluator
	ScatterSearchV1	Search using an evolutionary scatter search algorithm
	SubsetSizeForwardSelection	Extension of LinearForwardSelection that performs an internal cross-validation in order to determine the optimal subset size
Ranking Method	**Ranker**	Rank individual attributes (not subsets) according to their evaluation

To illustrate the attribute selection activity, the open source data set already used in the previous section has been analyzed, applying the *CfsSubsetEval* evaluation and *BestFirst* search methods (Figure 4.14). The attribute that has to be treated as a class also has to be identified.

```
=== Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval
Search:weka.attributeSelection.BestFirst -D 1 -N 5
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Evaluation mode:evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
      Best first.
      Start set: no attributes
      Search direction: forward
      Stale search after 5 node expansions
      Total number of subsets evaluated: 12
      Merit of best subset found:    0.887

Attribute Subset Evaluator (supervised, Class (nominal): 5 class):
      CFS Subset Evaluator
      Including locally predictive attributes

Selected attributes: 3,4 : 2
                 petallength
                 petalwidth
```

Run information gives the following information:
- used evaluator : *CfsSubsetEval*
- search method : *BestFirst*
- relation name (i.e. name of the set) : iris
- number of instances : 150
- number of attributes : 5 and the list of attributes
- evaluation mode : evaluate on all training data
- starting set : no attributes
- search direction : forward selection.
- search stops after 5 node expansions
- total number of subsets evaluated : 12
- merit of the best subset : 0.887

The attribute evaluator : CFS Subset Evaluator

At the end, the algorithm has selected the following attributes: petallength and petalwidth.

Figure 4.14: Results of Attribute Selection

4.4.2 Multi-label classification

In the traditional task of single-label classification, each instance is associated with a single-class label. When each instance may be associated with multiple labels, this is known as multi-label classification. Although single-label classification is considered the standard task, multi-label classification is by no means less natural or less intuitive. The human brain can naturally associate one idea with multiple concepts. The use of multiple labels implies an extra dimension that affects both the learning and evaluation processes. In this case, the evaluation process is no longer straightforward, since a simple correct/incorrect evaluation no longer suffices to convey the comparative predictive power of a given classifier. Thus, different evaluation methods are needed.

Learning is affected by label correlations, or label relationships, that occur in the multi-label dimension. The issue of label correlations directly influences a further issue: computational complexity. Instead of choosing a single class label from a label set, a multi-label classifier must consider combinations of labels.

Problem transformation is the process whereby a multi-label problem is transformed into one or more single-label problems. This approach is discussed in (Tsoumakas & Katakis, Multi-Label Classification: An Overview, 2007). The prime advantage of problem transformation is flexibility. Depending on the context, some classifiers may demonstrate better per-

formance than others, whereas the algorithm adaptation methods are usually designed with a specific domain in mind. In the following part, the fundamental problem transformation methods which are widely used throughout the literature are listed (Read, PhD thesis: Scalable Multi-label Classification, 2010):

Binary Relevance method (BR)

Pairwise Classification method (PW)

Label Combination method (LC)

Ranking and Threshold method (RT)

4.4.3   Multi-dimensional classification

Multi-dimensional classification (MDC) is a supervised learning problem where an instance is associated with multi-class labels, rather than a single-class label, as in traditional classification (Read, Martino, & Luengo, Efficient monte carlo methods for multi-dimensional learning with classifier chains, 2014). In traditional classification, each instance of the dataset D is associated with single-class label and then a classifier is trained to predict the class of a new unclassified instance. In real world application, it is very common to assign much different kind of judgments to a same instance (multiple labels). In addition, from perception point of view, the human brain can naturally associate different 'levels' (multiple classes) of an appropriate judgment to each instance depending on the personal impression. Thus, multi-dimensional classification is considered as a very natural and intuitive classification task that has very wide variety of research domains, such as image classification (Boutell, Luo, Shen, & Brown, 2007) (Qi, Hua, Rui, Tang, & Zhang, 2009), information retrieval and text categorization (Zhang & Zhou, Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization, 2006), automated detection of emotion in music (Trohidis, Tsoumakas, Kalliris, & Vlahavas, 2008) or bioinformatics (Zhang & Zhou, Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization, 2006) (Barutcuoglu, Schapire, & Olga, 2006).

Although MDC is considered as very intuitive classification task, at the same time, this classification task is a more difficult problem than the single-class case. The main problem is that there is a large number of possible class label combination, a corresponding sparseness of available data, and inability of straightforward application of basic single-class learning techniques. In general, the training dataset $D = (X, Y)$ consists of X, the set of instances and Y, the set of labels (Figure 4.15). $x = R^M$ is a M-dimensional attribute space where $x \in X$ is an attribute vector that represents the instance, $x = [x_1, x_2, …, x_i, …, x_M]$ and $X = [X^{(1)}; X^{(2)}; …. ; X^{(N)}]$, $X^{(N)}$ is N x M input matrix. $y = N^L$ is a L-dimensional label space where $y \in Y$ is a label vec-

tor, $y = [y_1, y_2,..., y_j,...., y_L]$ and $Y = [Y^{(1)}; Y^{(2)}; .... ; Y^{(N)}]$, $Y^{(L)}$ is N x L output matrix. Each label $y_j = \{1,..., K\}$ can have K classes, so depending on the number of labels L and classes K in a label, different classification problems can be identified. For instance, if L = 1 and K = 1, then this classification problem represents the single-label classification. If K > 2, then that is single-label multi-class classification. The latter classification problem can be easily transformed into a multi-label binary classification where each different class will be introduced as separate label in a binary way, representing its relevance. This way, the multi-label problem is turned into a series of standard binary classification problem that can be solved with any off-the-self binary classification. Having multiple labels, the problem of training classifier preserving the relations between labels arises. Finally, if L > 1 and K > 2, this represents the case of the most general classification problem and MLC is considered as a subset of multi-dimensional classification.

Unlike solving multi-label learning problems, for solving multi-dimensional learning problems, there are two general approaches to training classifiers: Networks Classifiers and Chain Classifiers. The first approach consists of application of most widely used Networks Classifiers such as Neural Networks (NN) and Bayesian Networks (BN). The second approach consists of techniques that can be applied to train multi-dimensional classifiers and they can be split in two groups such as: Independent Classifiers (IS) and Classifier Chains (CC). On the other hand, with changing the basic learning algorithm but preserving the same structure, Classifier Chains has many of its improved versions such as: Monte-Carlo Classifier Chains (MCC) (Read, Martino, & Luengo, Efficient monte carlo methods for multi-dimensional learning with classifier chains, 2014), Bayesian Chain Classifiers (BCC) (Zaragoza, Sucar, Morales, Bielza, & Larranaga, 2011).
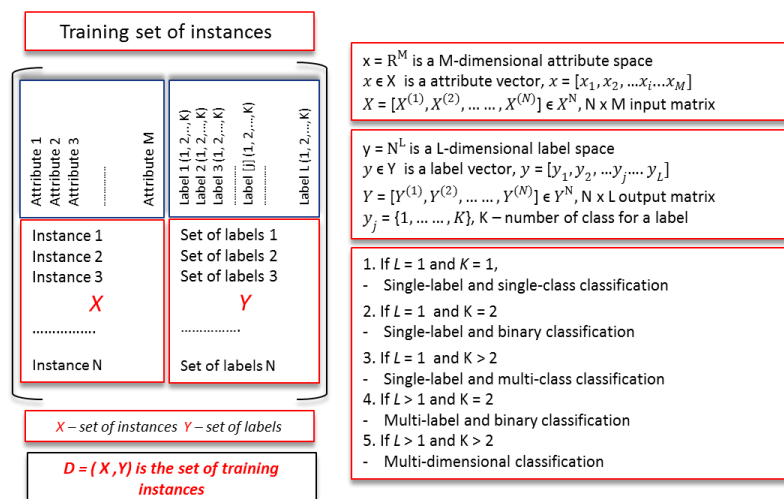

Figure 4.15: Training set of instances

4.5   Applications in various domains

Machine Learning Techniques have been widely used to discover structural patterns for analyzing problems in medical and biology application (Lemm, Blankertz, Dickhaus, & Müller, 2011), (Maddouri & Elloumi, 2002), (Summmers & Wang, 2012), (Wang, Nie, & Lu, 2014), (Wasan, Uttamchandani, Moochhala, & Yap, 2013). Next, wider applications of MLTs are in prediction of stock market (Lee, 2009), (Luo & Chen, 2013), (Ni, Ni, & Gao, 2011), where the idea is to discover which are the most relevant factors (features selection) that will influence the stock market. Furthermore, the most common application of MLTs is in image processing (Abreu & Fairhurst, 2008), (Conilione & Wang, 2011), (Lattner, Miene, & Herzog, 2004), (Motaal, El-Gayar, & Osman, 2010), (Negri, Dutra, & Sant-Anna, 2014), (Sajn & Kukar, 2011), (Schwenker & Trentin, 2014), where machine learning is used to discover classification patterns for feature recognition and automatic annotation. There are systems that are built to interpret the perceived object or interpret an image. However, these systems do not assume understanding the shape of the perceived object or its surroundings. The term "shape understanding" has a range of meanings, but in general, shape understanding refers to a computational, information processing approach to shape interpretation. The shape understanding denotes and interdisciplinary research area that includes data processing, statistical and syntactic pattern recognition, artificial intelligence (MLTs), and psychology. In the literature, the term "shape" often refers to the geometry of an object's physical surfaces which can be in 2D or 3D geometric space. The investigation for classification patterns of the 2D shapes for features recognition is a research topic covered by image processing. Today, the representation of 3D objects is not common only in industrial and engineering design, but also, the 3D shape representations are often used in games, medicine, and archaeology research fields. The widespread implementation of the 3D geometric representation of shapes in different domains requires their greater integration and creating tools for automatic 3D shape classification and retrieval. The existing classification and retrieval techniques are more oriented to text and features recognition on 2D shapes, which cannot be directly extended to 3D shapes (Atmosukarto, 2010). There are some works related to 3D models classification and retrieval (desJardins, Eaton, & Wagstaff, 2006), (Ip & Regli, 2005). The latter paper discusses the possibility of discovering classification patterns for shape description of already created 3D CAD models, using supervised learning techniques. An efficient algorithm for 3D shape classification and retrieval requires: 1. 3D shape representation suitable for the search techniques and 2. effective similarities function for computing the distances between entities in the feature space (Laga, 2009). The features of the shape are often of different scales, which mean that incorporating them directly into the classification model without normalizing them will result in low classification performance. Therefore, a pre-processing step is required in order to transform them in a same scale (same dimension or dimensionless). The application of MLTs is widely spread in solving many supervised learning problems, allowing the automatic selection of relevant attribute of a single 3D model within a class of shapes. One of the basic learning approaches is the k-Nearest Neighbor

(kNN) classifier. It has been used for the classification of 3D protein databases (Ankerst, Kastenmoller, Kriegel, & Seidl, 1999), and also 3D engineering part (Ip, Regli, Sieger, & Shokoufandeh, 2003). Other learning approach presented for the first time by Vapnik (Vladimir, 1995) is Support Vector Machines (SVM). Hou et al. (Hou, Lou, & Romani, 2005) introduced a supervised semantic classification method based on Support Vector Machines (SVM) to organize 3D models semantically. Later, Hou and Romani (Hou & Ramani, 2006) combine both semantic concepts and visual content in a unified framework, using a proba-bility-based classifier. Xu et al. (Xu & Li, 2007) illustrate how to select the training set of in-stance and how to setup the parameters of the training model of a Neural Network (NN) algorithm for learning on 3D unclosed polygonal models. In a design process, designers need tools to help them understand custumers' needs and thereby predict their appreciation level of a new product. In order to do that, Walid et al. showed that Bayesian Networks (BN) are a very flexible and powerful methods in preliminary perceptual designs in terms of simulation and prediction capacities (Walid & Yannou, 2007). Many researchers have been using the learning algorithms in different applications in order to map semantic to the 3D geometric shapes. They all agree that there is no 'best' learning method, but which of the learning method is better for a given application. It is not evident which learning algorithm is the best in mapping semantics to 3D shape and therefore, if we want to apply MLT approach in map-ping semantics to shapes, we have to consider several learning methods.

Danglade et al. (Danglade & Veron, 2014) has introduced the use of MLTs when pre-paring CAD models for FE simulations. This paper presents the way how MLTs can be used to learn to avoid the adaptation of CAD models for simulation models which is considered as a particular step of disfeaturing (identification of features to be deleted or retained). In this case, the inputs are CAD models as well as processes describing what it needs to be done to prepare the FE simulation model. In design application, there are various works aiming at understanding the relation between the emotion evoked and the product shape given on image (i.e. the most appropriate description such as calm, feminine, aggressive), from cus-tomer interviews (Lesot, Bouchard, Detyniecki, & Omhover, 2010), (Lu, et al., 2012), (Ren, 2012). There are only very few works trying to find the shape parameters that can be directly applicable for creating high-level manipulation tools used in the early phases of the product design process (Giannini & Monti, 2010). Giannini et al. in the paper (Giannini, Monti, & Podehl, Aesthetic-driven tools for industrial design, 2012) present the possibility of mapping relationships between shape (free-form curves) and its aesthetic character.

Xu et al. in the paper (Xu, Kim, Huang, & Kalogerakis, 2015) present a general overview of the entire data-driven 3D shape analysis and processing process concept (Figure 4.16). This concept consists of three layers: 1. Data collection, 2. Data management, and 3. Data analysis and processing. There are two major ways of 3D data generation, 3D sensing and 3D content creation. The 3D database models are sparsely enhanced with segmentation and labelling (classification), in order to support data-driven shape analysis and processing sup-ported by machine learning techniques.
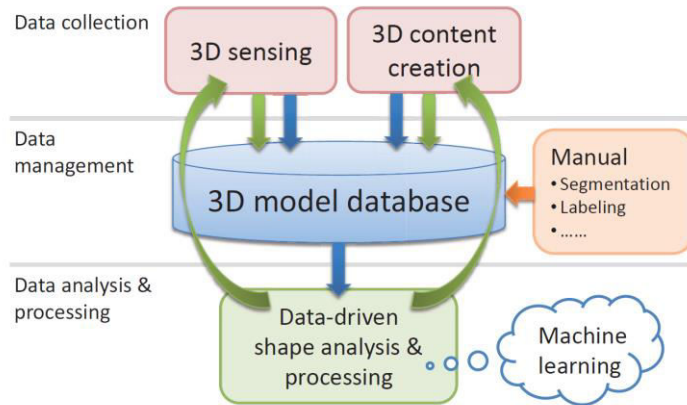
Such learned knowledge, can later be returned to the Data collection layer enriching the 3D content with semantics. Such 3D data with semantic information can be included into the database to enrich it and facilitate further data-driven applications.

4.6   Conclusion and Synthesis

In this chapter, the basics of the Machine Learning Techniques (MLTs) have been presented. The MLTs, as core subarea of the Artificial Intelligence (AI), are most widely used methods for solving Data Mining (DM) problems. In general, the tasks that are addressed by the Data Mining problems are divided in two groups: Predictive tasks and Descriptive tasks (Figure 4.5). Thus, the Machine Learning focuses on the predictive tasks, learning from past experiences, whereas the descriptive DM problems are solved using different approaches depending on the task. The MLTs provides very efficient algorithms to deal with large variety of learning problems which can be grouped in two major groups of learning problems: Supervised and Unsupervised learning. The supervised learning uses learning algorithms that learns association between instances and the class labels which are assigned by someone. Since, the assignment of class is done by an expert (a person who has specific knowledge in a given field), the supervised learning can help us to extract knowledge from the supervisor and represent it in a form of rules. The second type, the unsupervised learning, is used when instances are not labeled and it aims at finding intrinsic relationships between instances. The objective of this thesis is to investigate the relationships between 3D free-form shapes and its aesthetic impression from a customer's point of view. Therefore, the investigation for finding whether there is a common judgment (rules) for the aesthetic appearance of shape is considered as supervised learning problem.

It is very important to emphasize that we do not intend to develop a new Machine Learning Techniques, but to apply the already existing well-known learning algorithms. Of

course, this work does not aim at verifying or testing all existing learning algorithms, but rather at investigating which of them are most suitable for our application. After analyzing the works done on testing learning algorithms in different application fields, the majority of the authors have identified the following algorithms to be the most performant: C4.5 decision tree (J48), IBk (k-NN), SMO (SVM), NaiveBayes (NB), and RIPPER (JRip). Therefore, these learning algorithms have been adopted to be tested in our application. From the analyzed works, it is very difficult to select which is the most appropriate in our application, because they are all more or less performant in a given domain. Additionally, performance of the learning algorithm significantly depends on the type of learning problems, as well as the type of the attribute and process depicted by the dataset (instances). For the current application, the instances are the shape of geometric entities (curves and surfaces), the attributes are geometric quantities (area, length, curvature and so on), and the assignment of class labels to all instances is done by interviewing non-professional designers. In addition, when investigating aesthetic perception of shapes, the selection of the set of shapes is also important and affects the entire classification process. These questions and many other must be addressed in the next section in order to be able to map aesthetic properties to 3D free-form shapes.

Classification framework specification and its validation on curves

This chapter introduces the proposed classification framework for mapping aesthetic properties to free-form shapes and its validation on curves. The chapter begins by presenting the overall framework and its constituting parts (Section 5.1). The next section (Section 5.2) describes the setup of the overall framework for testing the performance and validation on curves. This section explains the way the initial dataset of instances is generated. In the proposed approach, the generation of the instances starts by defining the space of shapes (Section 5.2.1), then by implementing deformation modes (Section 5.2.2), after which the entire initial dataset (IDS) of curves is being created. The attributes identification (Section 5.3) and classification (Section 5.4) of the entire IDS are then presented. After the brief listing of the considered learning methods (Section 5.5), the experimentations for validating the framework on curves are presented (Section 5.6). The experimentation activity involves modeling the measure of the straightness, preparing the dataset for applying the learning algorithms (Section 5.6.1). Further, this section presents the accuracy of the created classification models, using dimensional (Section 5.6.2) and dimensionless (Section 5.6.3) attributes, as well as the capability for selecting the most relevant attributes (Section 5.6.4). The last section (5.7) gives a synthesis of this chapter.

5 Classification framework specification and its validation on curves

To bridge the gap between the geometric definition of the shape and its aesthetic properties, an overall framework has been set up (Figure 5.1). It aims at identifying aesthetic classification rules while using Machine Learning Techniques (MLTs) on extracted geometric quantities characterizing the shape. The overall framework describes the main activities of how to investigate the aesthetic properties of shapes. Even if the final objective is to apply this framework on free-form surfaces, we decided to test and validate it in a similar application on free-form curves. Therefore, this chapter exposes the validation of the framework on curves and presents the results of the implementation. Actually, the validation on curves is possible due to the existence of a complete definition of the measure of straightness for curves and classification of curves in different classes for different ranges of the measure of straightness. This framework can be seen as a guided path for structuring and understanding aesthetic properties of shapes, and the results open new perspectives for creating high-level shape modification tools.
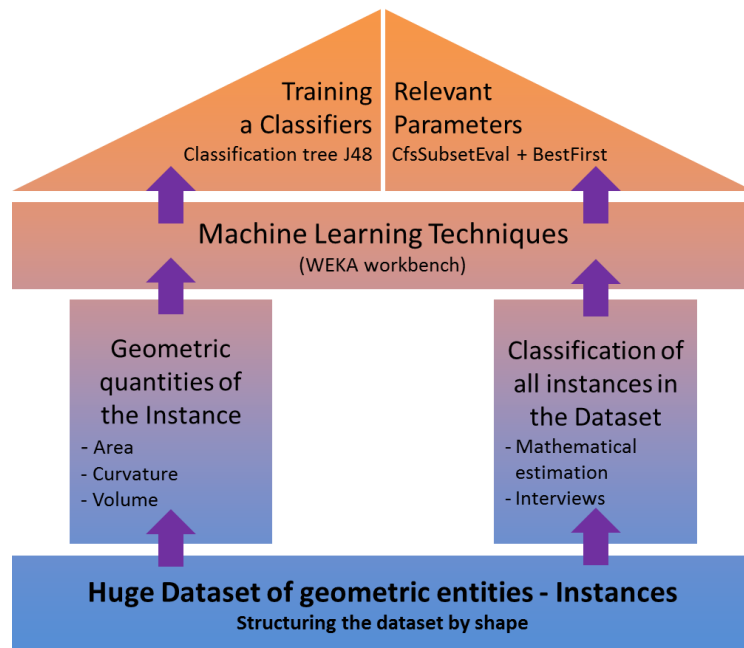
Figure 5.1: The overall framework

5.1 Overall framework

As introduced in the previous chapter, the use of MLTs requires the definition of a huge structured dataset on which classifiers will be trained. As a consequence, the proposed framework also relies on a huge structured dataset of geometric entities, i.e. free-form curves in this chapter, which form the base of the temple and are called instances (Figure 5.1). This is a crucial part of the framework because having a structured dataset of several thousands of instances improves the relevance of the extracted classification rules. Also, the choice of the selected shapes is very important. This may affect the identified classification rules if the variability of the shapes is limited and does not cover the possibilities of shape arrangements that may affect the perception of a given aesthetic property. Therefore, specific methods for the creation of those instances have been devised through the modification of instance replications (e.g. deformation modes and morphing between shapes). Furthermore, the approach followed for associating the classification to the single instance can be different and may affect the organization and number of the instances in the dataset. For instance, if the classification results from interviews in which users/designers classify each shape, an instance may be repeated to verify the consistency in the classification. On the contrary, if the assignment of classes is done automatically (e.g. by using mathematical estimation), then there is no need to repeat the same shapes at the beginning.

The second element of the framework is the pillar representing the classification of all the instances of the dataset. A class is assigned to each instance of the dataset. In this thesis, we have been developing two methods: 1) one using automatic assignment of a class based

on a mathematical estimation, and 2) the other while conducting interviews over a group of participants. The first method is adopted for verifying the overall framework. It is simpler and faster than the second one since it exploits a mathematical equation that computes a measure of the aesthetic property to be classified. At the end, the instance is classified according to whether this measure belongs to different class ranges whose limit values have been obtained through interviews carried out by a precedent work (Giannini, Monti, Pelletier, & Pernot, 2013). The second method is more complex and has to face several challenges inherent to the reliability of the classification (e.g. finding an efficient and intuitive way for conducting the interviews and finding people from different countries and different backgrounds willing to participate in the interviews).

The third element of the framework is the other pillar of the temple that gathers geometric quantities characterizing the instances (namely, the free-form curves in this chapter and surfaces in Chapter 6) that are potentially usable for the specification of the classification rules. Here, the key issue is to define which geometric quantities are relevant regarding a given aesthetic property. A very important point to be considered is how to combine the geometric quantities (e.g. area, curvature, length, volume and so on) in order to construct shape descriptors independent of the size, position and orientation of the considered shapes. The choice of the geometric quantities is crucial since the instances will be characterized and described by those values from which MLTs will try to extract the classification rules. Of course, if the quantities to be analyzed are not well chosen, the identified rules may be not representative. To get dimensionless shape descriptors, it is possible to define ratios between geometric quantities, or groups of geometric quantities. The term "dimensionless shape descriptors" define parameters without dimension. On one hand, the use of ratios helps transforming geometric quantities into dimensionless parameters. On the other hand, it suffers from lack of getting infinite values for the ratio if the denominator is null.

The fourth element of the framework is the beam of the temple that corresponds to the adopted Machine Learning Techniques (MLTs) and associated control parameters. Same as with a real temple, the beam is supported by the pillars. Here, it relies on both the classified instances and the associated geometric quantities. This part represents the actual application of the MLTs with the selection of the most suitable learning algorithms for this kind of application (i.e. link between geometric quantities and aesthetic properties). Here, the main challenge relies on the identification of the best couple of classifier and associated control parameters, i.e. the couple that would maximize the rate of well-classified instances. If the instances are classified with more than one label (multiple labeling), then, before applying the basic single-label learning algorithms, dedicated problem transformation methods have to be applied. Actually, such methods transform a multi-labeled classification into a single-labeled classification while preserving the relation between all labels. In the implemented version of the approach, five of the most widely used basic learning algorithms have been tested and tuned: C4.5 Decision Tree, Naïve Bayes, k-Nearest Neighbor, Support Vector Machine and Classification Rules.

The fifth element of the framework is the roof that represents the results, i.e. tuned classifiers working on a set or relevant parameters. First, different classifiers obtained with different methods are tested and parameterized so as to optimize the classification accuracy on known instances. Then, some tests are performed to predict the class of unclassified instances. For a given aesthetic property, the relevant attributes are also identified. This last element of the framework is somehow the final result of the proposed framework. It allows the identification of classification rules that can then be applied on new, unknown cases. These identified rules are also the first building blocks for the definition of higher-level modeling tools acting directly on aesthetic properties, rather than on low-level geometric quantities.

The different elements of the framework are detailed in the next sections.

5.2 Setting up of the framework

As previously mentioned, to verify the proposed framework and the classification efficiency of the state-of-the-art learning algorithms in our context, we applied them on the specific case of curve classification, with respect to five different classes of straightness.

The input to a Machine Learning Technique is a set of instances characterized by numerical values, namely the attributes as described in section 5.3. Even if each instance contains certain important and implicit information, and then, all the information that might be useful for the learning step has to be explicitly specified. By collecting as many instances as possible from the same type of geometric entity (curves or surfaces, but not a mix of these two types), MLTs are capable to extract the common classification patterns or relevant attributes. From the computational point of view, the instances of the geometric entity can be provided as an IGES file from which the geometric quantities can be computed. Each instance is an individual of the concept to be learned and it is characterized by the values of a set of predetermined attributes. Considering the aesthetic – shape relations discovering, the instance is a geometric model (curve here, or surface in Chapter 6), and the shape is the holder of the aesthetic properties that are implicitly correlated to the shape of the geometric model. Therefore, if we want to explore the aesthetic property of free-form shapes, it is necessary to create a huge set of geometric instances. To achieve this objective, we decided to consider the ones obtained at different steps of the morphing of a given curve (later surface) to a target one. The set of these target shapes is indicated as space of shapes in the following discussion. Theoretically, there exists an infinite variety of free-form shapes and all of them could be applicable to a real product. However, not all those shapes can be considered and it is therefore important to identify the most meaningful ones regarding their application (appearance). Furthermore, the set is meaningful if it includes shapes presenting important variations on their behavior (e.g. presence/absence of inflection points,
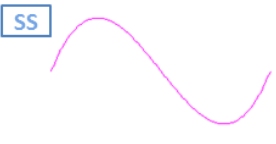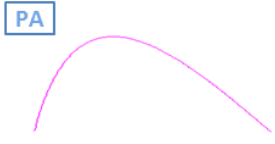
a/symmetry). The signification of the term "meaningful" is whether the given shape can be used, or it exists in a real product. For instance, the top of a coffee machine or a car back cannot be too much bumped or with sharped edges. It is important to notice that the generated rules will of course be strongly dependent on the set of instances used as inputs. Most probably, those rules will have a validity domain that will be related to a given application. This space of shape is designed in such a way that every single shape, in our opinion, is interesting in terms of generating specific aesthetic impression (character). Actually, in case of free-form curves, we do not examine the straightness of curves, but we test the capability of the MLTs embedded in the proposed framework. So, the selection of free-form shape is less relevant but anyway, they have to be meaningful. Finally, after the definition of the space of shapes, a huge dataset of instances is created by application of instances replication methods (deformation modes).

The next subsections develop on the space of shapes that is to be used and to be followed by the generation of the dataset.

5.2.1 Space of shapes

A starting point for the creation of the dataset is the definition of the space of shapes (set of target shapes). The definition of the space of shapes is an inevitable activity in order to reduce the entire set of possible shapes to very few shapes. By selecting different shapes, we want to investigate the influence of different shape characteristics to the perception of a given aesthetic property. Therefore, a space of shapes has been created, aiming at verifying the learning capability of the MLTs for different shapes. In order to do that, three different shape classes have been considered: parabolic, sinus and elliptical (Table 5.1), and each of them is given in two variations: symmetric and asymmetric.

Table 5.1: Space of shapes

Space of shapes	Parabolic (P)	Elliptical (E)	Sinus (S)
Symmetrical (S)	PS	ES	SS
Asymmetrical (A)	PA	EA	SA

90

It is important to underline that all shapes in the space of shapes are created using cubic Bézier curves which are a special case of the NURBS curves – four control points and weights equal to one. The motivation for selecting these shapes is that they are widely used for creating more complex shapes. In addition to this, to validate the framework, the selection of the target shapes is less relevant than testing the capabilities of the MLTs. The parabolic shape has been used to represent bumped shapes, while sinus kinds of shapes are used to represent the undulated shapes with the occurrence of inflection points. The elliptical shapes are representing the bumped shapes that have higher curvature deviation. They/ This aim/ aims at investigating the correlation of the curvature distribution and variation with the considered aesthetic property, i.e. the straightness. Furthermore, we also want to investigate the capability of the MLTs to learn from shapes that are asymmetric, so previously mentioned shapes (parabolic, sinus and elliptical) are modified to obtain also asymmetric shapes. Finally, the space of shapes consists of six categories of target shapes, and the three basic categories are given in two variations (symmetric and asymmetric).

5.2.2 Dataset of curves

Guided by the main requirement of Data Mining methodology, a huge dataset of instances is created through the application of deformation models from initial straight lines to the six previously introduced types of curves (target shapes of Table 5.1). For this purpose, a deformation model (Figure 5.2) has been defined and applied on an initial straight line to create many 2D free-form curves. The adopted deformation model uses three deformation modes (ε_1, ε_2 and ε_3) based on two displacement vectors (blue and yellow), applied on two middle control points, whereas the other two end points of the parametric curve remain unchanged. Using each of the three deformation modes (ε_1, ε_2 and ε_3), sets of curves corresponding to the three types (parabolic, elliptical, and sinus) of curves in the space of shape are generated (Table 5.1). Then, 200 displacements are assigned to each displacement vector which also rotates under given angle in clockwise or counterclockwise direction, depending on which target shapes (Table 5.1) are to be reached. The motivation for applying the deformation modes and the gradual modification of 200 displacements is to explore, as accurately as possible, the entire domain of possible shapes of curves. The small gradual deformation aims at identifying when a small change of the shape changes the class of the curve. Having a set of curves generated, using small gradual deformation, will help the learning algorithms to train the classifier more accurately, which will improve its classification accuracy. Figure 5.2 illustrates the vector directions (blue and yellow arrows), the vector rotation (curved arrows), and the resulting types of shapes. The solid arrows represent displacement vectors in their initial positions. They rotate a number of times (e.g. x11 or x6 for the curved arrows) under given angle until they reach the final positions (dashed straight arrows).
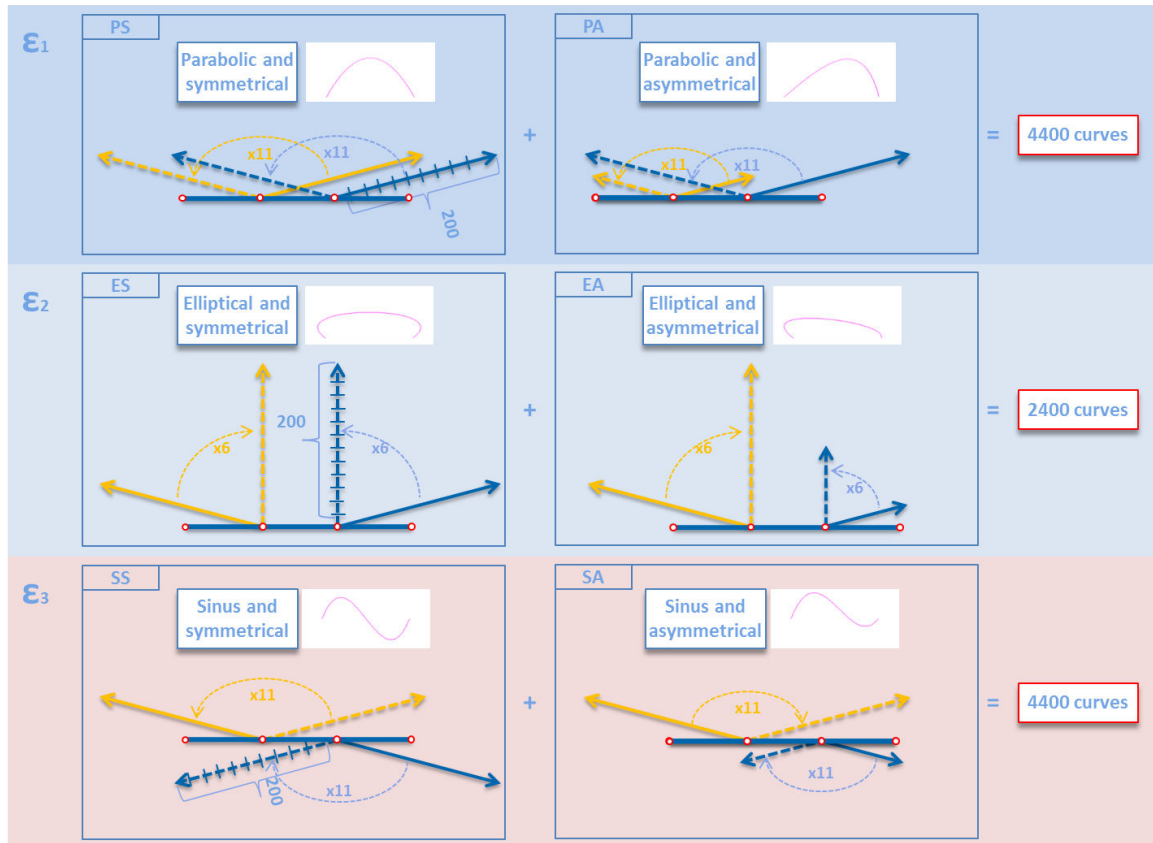
Figure 5.2: The deformation model used to generate a set containing a variety of 2D curves

For example, in the ε_1 mode, in the first subset (PS of Figure 5.2), we selected 11 different directions and then both vectors are equal and parallel with the 11 vector directions and 200 positions for each direction. The displacement vector directions have been selected in such a way that the entire range from 0 to 180 degrees was covered through their uniform distribution. The initial position of the displacement vectors is given by the solid straight arrows. Then, they rotate 11 times until they reach the final position that is represented by the dashed straight arrows. In the second subset (PA of Figure 5.2), the displacement vectors are still parallel, but only their modules (lengths) are different (blue is too times the yellow), aiming at obtaining asymmetric shapes. The same principle is adopted for the ε_2 (ES and EA) and ε_3 (SS and SA) deformation modes with different number of vector directions. In these two cases we provided 6 different initial directions. To summarize, the deformation modes allow to create: with ε_1 (11 directions * 200 positions * 2 subsets) = 4400 curves, with ε_2 (6 directions * 200 positions * 2 subsets) = 2400 curves, and with ε_3 4400 curves, too. Collecting the curves obtained with the above described deformation modes gives us the set of 11200 curves applied on four different straight initial curve lengths (5, 8, 9 and 12 cm) to produce the global set ST of 44800 2D curves, as illustrated in Figure 5.3.
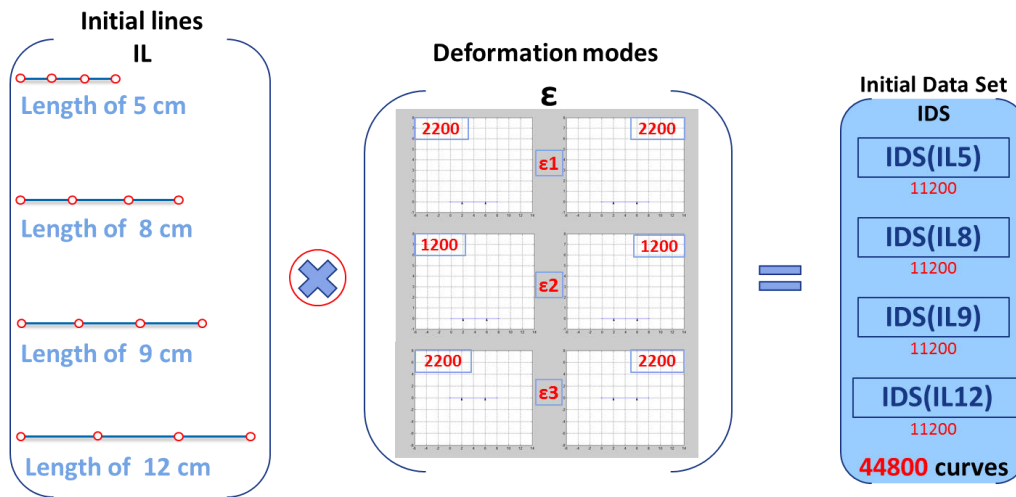
Figure 5.3: Generating a set of 2D curves from initially straight lines applying three different deformation modes

5.3 Attributes

Each instance that provides the input to machine learning techniques is characterized by its values with respect to a predefined set of **attributes**. A dataset used for practical data mining is represented as a matrix of instances versus attributes, where the instances are the rows while the attributes are the columns. The value of an attribute for a particular instance is a quantification of the referred attributes. Generally, there are two different types of attributes: *numeric* and *nominal*. Numeric attributes are given as numbers, i.e. using either real or integer values. Nominal attributes have values that are concrete symbols and serve just as labels or names, hence the term nominal.

Concerning the curves, the following Figure 5.4 represents the geometric quantities that have been identified for a curve. Since a curve is a geometrical entity, its characterization can be done by representing the curve through its intrinsic geometric quantities like its length (L), area (A), curvature (C) and cord length (l). These parameters have first been used in the revised measure of straightness (Giannini, Montani, Monti, & Pernot, 2011). Therefore, these geometric quantities of a curve are considered as attributes that have been further used in the application of the MLTs.
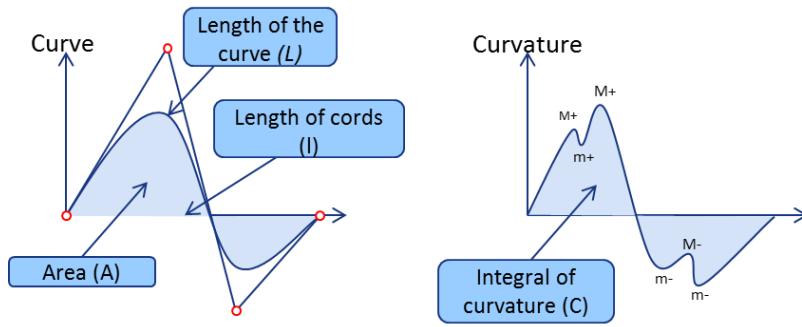
Figure 5.4: Geometric quantities of a curve

The curve is given as a function of the parameter "u":

$$P(u) = (x(u), y(u)), \ u \in [0,1] \quad \dots\dots\dots\dots\dots\dots\dots \ (5.1)$$

$$C = \int_0^1 |k(u)| \, du \quad \dots\dots\dots\dots\dots\dots\dots\dots \ (5.2)$$

is the integral of the absolute value of the curvature $k(u)$

$$A = \int_0^1 y(u) \cdot \dot{x}(u) \, du \quad \dots\dots\dots\dots\dots\dots\dots \ (5.3)$$

is the value of the area between the curve and the line that joins the two extremes of the curves.

$$L = \int_0^1 \sqrt{\dot{x}(u)^2 + \dot{y}(u)^2} \, du \quad \dots\dots\dots\dots\dots\dots \ (5.4)$$

is the length of the curve

$$l = \sqrt{(x(1) - x(0))^2 + (y(1) - y(0))^2} \quad \dots\dots\dots\dots \ (5.5)$$

is the length of the cord between the two end points of the curve.

5.4   Classification

As previously said, for testing the framework we used the classification obtained with the interviews from Giannini et al. (Giannini, Monti, Pelletier, & Pernot, 2013) that used the measure (eq. 3.3). This measure of straightness represents the character of the curves better than the initial one proposed by the FIORES II Project. From the work in (Giannini, Monti, Pelletier, & Pernot, 2013), it has been concluded that there is a general convergence in the categorization of curves according to very not, not, fairly, and very judgments for the straightness property (see Table 5.2). Therefore, in this work, the measure of straightness

94

expressed with the equation and the classification given in Table 5.2 are used in the next analysis to create the model for straightness category prediction.

Table 5.2: Curve classification with respect to the straightness

Straightness - S	Class
(0, 0.7)	very-not-straight(nS)
[0.7, 0.9)	not-straight(ns)
[0.9, 0.98)	fairly-straight(s)
[0.98, 0.999)	very-straight(S)
[0.999, 1]	straight-line(SS)

5.5 Considered learning methods

As previously described, in terms of investigation of the classification patterns (the first data mining task), there are three ways of using MLTs. The first one is to apply a learning method to a dataset and analyze its output in order to learn more about the data. The second way is to use a learned model in order to generate prediction on new unclassified instances. The third way is to apply several different learners to compare their performance in order to choose the most suitable for a given application. As explained in the previous chapter, the comparison for selecting the most suitable learning algorithms in our application is done between five of the most widely used state-of-the-art learning algorithms:

1. *Classification trees J48 (C4.5)*
2. *Naïve Bayes (NaiveBayes)*
3. *K-Nearest Neighbor (IBk)*
4. *Support Vector Machines (SMO)*
5. *Classification rules (RIPPER)*

Regarding the second data-mining task (Attribute Selection), there are tools to identify which attributes are relevant with respect to a certain property, and which attributes can be omitted without affecting the results of the analysis. In order to verify the capability of Attribute Selection, two pairs of evaluation algorithms and searching methods, described in the previous chapter, have been applied:

➢ CfsSubsetEval + BestFirst
➢ InfoGainAttributeEval + Ranker

5.6 Experimentations

5.6.1 Modeling of the straightness's rules identification problem

Aiming at modeling the straightness, the measure expressed by the equation (3.3) is adopted as a mathematical model of curve straightness, and its parameters (C, A, L, l) represent the curve's geometric quantities. Therefore, to create the model for straightness class prediction, the curve's quantities and the measures of straightness for the entire dataset of curves – IDS (Figure 5.3) have to be computed. In order to get the data, a program has been developed in Matlab. It includes a function for the automatic generation of curves and a function that computes curve's quantities and its associated curve's straightness class according to the equation 3.3 and Table 5.2. At the end of this Matlab program, a so-called ARFF file is generated, which is the format adopted by WEKA (Waikato). It contains two distinct sections: the first section is the *Header* information that is followed by the *Data* information (Figure 5.5). The Header contains the name of the relation (@RELATION), a list of the associated attributes (@ATTRIBUTE) with their types and, if necessary, comment lines starting with the character '%'. The Data section contains the data declaration (@DATA) line followed by the instance lines. In the first column in the data section are given the values for the different curve length (L), the second column represents the values for the different curve Area (A), and in the last column, the classes for each curve instance are given. The Data section contains a line for each instance of curves with the computed values of the curve's quantities and the corresponding class according to Table 5.2.

```
% Header section
@RELATION straightness

@ATTRIBUTE Length(L)      REAL
@ATTRIBUTE Area(A)        REAL
@ATTRIBUTE Cord(l)        REAL
@ATTRIBUTE Curvature(C)   REAL
@ATTRIBUTE class          {straight-line(SS),very-straight(S),fairly-straight(s),not-straight(ns),very-not-straight(nS)}

% Data section
@DATA
8.000000,0.000000,8.000000,0.000000,straight-line(SS)
8.003050,0.504172,8.000000,0.016272,very-straight(S)
8.004159,0.588237,8.000000,0.019139,fairly-straight(s)
8.005444,0.672313,8.000000,0.022079,not-straight(ns)
8.006906,0.756398,8.000000,0.025103,very-not-straight(nS)
```

Figure 5.5: Modeling of the straightness in WEKA

5.6.2 Classification using dimensional attributes

In terms of using machine learning algorithms, many different classification algorithms have been used in WEKA, but only the classification trees (particularly, tree J48) appeared to be the most convenient in the case of numerical attributes (Table 5.3). The classification tree J48 in WEKA is based on the C4.5 learning algorithm. Classification of instances that have dimensional attributes refers to those instances whose attributes are not only single values, but they are also accompanied with dimension. In our case, the dimension of the attributes is meter (L, l), meters squared (A) and meter to the minus one (C), whereas in other applications, the dimension of the attribute can be meter cubic (volumetric attribute), meter per second to the minus one (speed), and other geometrical and physical magnitudes.

The process of classification model creation consists of two steps: modeling detection and model evaluation. In the first step, the classification algorithm is applied on the input training set (TRS) which is a subset of the initial data set (IDS) of 2D curves. In the second step, a subset TES (10-30%) of the IDS (Figure 5.6) is only used to evaluate/test the model, giving the percentage of correctly and incorrectly classified instances. The division of the initial dataset IDS in a given portion of training set (TRS) and testing set (TES) depends on which splitting algorithms it has been using (Figure 5.6). For the evaluation process, there are three different ways: cross-validation, split test set and supplied test set. The use of those strategies has already been discussed in the previous chapter (Chapter 4). The first two methods are used for the classification model evaluation, while the third one is used for the validation of the classification model.
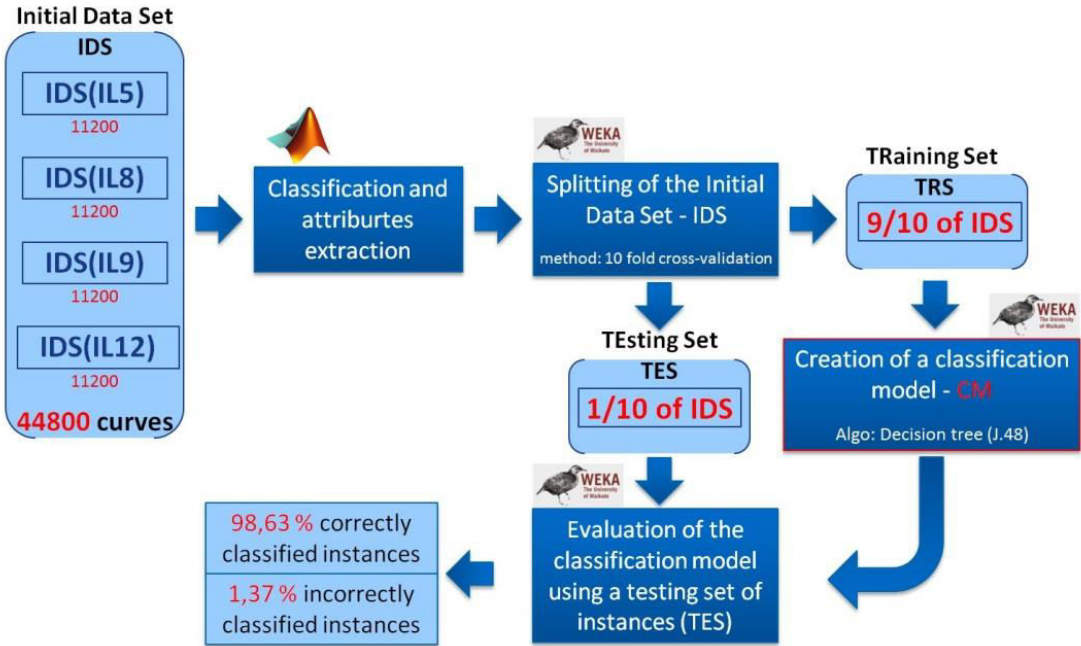

Figure 5.6: Evaluation process - 10 fold cross-validation

97

Results:

When the modeling process is over, learning schemes are automatically generated and, depending on which evaluation method has been used, we obtained a percentage of correctly classified instances varying from 98.63 % (for the cross validation) - to 98.18 % (for the split test). The classification accuracy of the various classification algorithms is given in Table 5.3:

Table 5.3: Accuracy of the classification algorithms with respect to the test modes

<table>
<tr><td rowspan="2"></td><td colspan="6" align="center">Classification algorithms</td></tr>
<tr><td>Test mode</td><td>C4.5</td><td>IBk</td><td>RIPPER</td><td>SMO</td><td>NaiveBayes</td></tr>
<tr><td rowspan="2">Correctly classified instances (%)</td><td>10 fold cross-validation</td><td>98,63</td><td>98,56</td><td>98,15</td><td>85,58</td><td>64,48</td></tr>
<tr><td>Percentage split − 66%</td><td>98,18</td><td>98,22</td><td>97,71</td><td>85,39</td><td>64,81</td></tr>
</table>

Moreover, when analysing the incorrectly classified instances, we can observe that they are classified just in the neighbouring classification class, and the error of the classification is not more than 0.5 %. The confusion matrix (Figure 5.7) shows the distribution of the classification to each class and measures the performance of the classification problem. For instance, from 5209 very-straight curves, using the classification model, 5119 curves are correctly classifed very-straight curves, whereas 21 and 69 curves are incorrectly classifed straight-line or fairly-straight. Since those two classes are neighbouring classification classes and this repeats to other classes of curves, the error is limited (Figure 5.7).

```
=== Confusion Matrix ===

    a     b     c     d     e    <-- classified as
 1927    21     0     0     0 |    a = straight-line(SS)
   21  5119    69     0     0 |    b = very-straight(S)
    0    50  6575   100     0 |    c = fairly-straight(s)
    0     0    75 11405   149 |    d = not-straight(ns)
    0     0     0   127 19162 |    e = very-not-straight(nS)
```

Figure 5.7: Confusion matrix

Having these results into account and the fact that the intervals of the classification class (Table 5.2) are very strict from a numerical value point of view, a change of 0.5 % in the

intervals will not make any visible changes in the appearance of curves, which allows us to consider that this classification model is relevant. Before training the classifier, some input parameter such as confidence factor (flag –C) and minimum number of instances (flag –M) have to be set up. The confidence factor is used to define the pruning (the smaller values incur more pruning), whereas the latter parameter gives the minimum number of instances per leaf. The output of the trained classifier is given in the form of classification tree which consists of number of leaves and the size of the tree. The classification tree CM (Figure 5.6) has 436 leaves and the size of the tree is 871 (double number of leaves minus one). The entire set of information that the classification tree provides has been explained in the previous chapter, so in the following two figures (Figure 5.8 and Figure 5.9) the outputs of our classification model CM are given when all the instances are used as inputs:

```
=== Summary ===

Correctly Classified Instances       44188               98.6339 %
Incorrectly Classified Instances       612                1.3661 %
Kappa statistic                          0.9807
Mean absolute error                      0.0063
Root mean squared error                  0.0728
Relative absolute error                  2.2175 %
Root relative squared error             19.3222 %
Coverage of cases (0.95 level)          98.8192 %
Mean rel. region size (0.95 level)      20.2437 %
Total Number of Instances            44800
```

Figure 5.8: Summary - Estimation of the predictive performance

The first figure (Figure 5.8) lists the main information related to the predictive performance. The CM classification model has correctly classified 98.63% of the instances while only 1.37% of the instances have been classified incorrectly. Further in this figure, the information related to statistics and classification errors are given.

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.989	0.000	0.989	0.989	0.989	0.989	0.996	0.980	straight-line (SS)
	0.983	0.002	0.986	0.983	0.985	0.982	0.994	0.979	very-straight (S)
	0.978	0.004	0.979	0.978	0.978	0.974	0.992	0.972	fairly-straight (s)
	0.981	0.007	0.980	0.981	0.981	0.974	0.991	0.976	not-straight (ns)
	0.993	0.006	0.992	0.993	0.993	0.987	0.996	0.992	very-not-straight (nS)
Weighted Avg.	0.986	0.005	0.986	0.986	0.986	0.981	0.994	0.983	

Figure 5.9: Detailed accuracy by class

The second figure (Figure 5.9) details the classification performance of the corresponding per class breakdown of the classification model. The *recall* values present the ratios between correctly classified instances and the total number of instances of a same class. The *precision* values present the ratios between the correctly classified instances and the total number of instances classified in a same class. The *ROC Area* is the area under the *ROC Curve* which is plotting of *FP Rate* (on the x axis) against *TP Rate* (on the y axis) for the entire training set.

Discussion:

One important aspect is that the classification in the curve category should be somehow independent of the curve dimension, i.e. it should reflect only shape behavior characteristics. Thus, to verify whether this model is independent from the initial line length and uses it to predict the straightness class of any curve, an evaluation of the model has been performed using curves obtained from other initial lines that have different lengths never learned, i.e. 0.8 and 80 cm. The results show that the model correctly classifies from **21.14 %** to **28.42 %** of all data instances only (Figure 5.10). So it demonstrates that this way of model creation does not guarantee to be size independent. Therefore, another approach for model creation has to be defined, one that will be more general and independent from the initial curves or attributes. In the next section, this new approach is presented together with the results obtained.
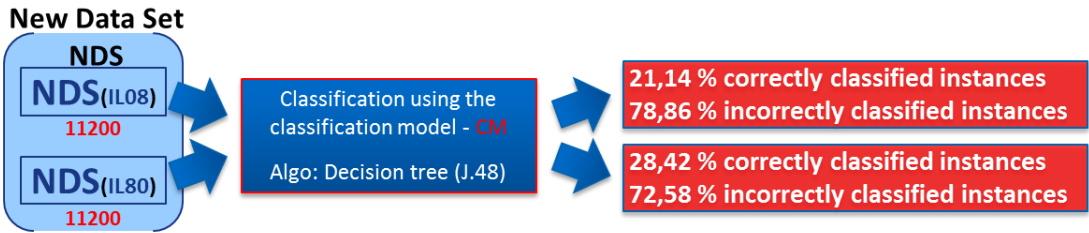
Figure 5.10: Validation of the classification model CM

5.6.3 Classification using dimensionless attributes

As mentioned before, a new approach has been explored in order to overcome the lack of low rate of correctly classified instances when using attributes that have dimensions. After an analysis of the mathematical model of the non-straightness (eq. 3.2), it was concluded that whilst the final measure is independent of the curve size, each single parameter (attribute) in the equation is related to the curve dimension. Thus, this model gives good

results in classification only for the range of curves used for its creation. This remains true for any other model and it has a validity domain that has to be given. Of course, if dimensionless attributes have been used to create the model, the validity domain may have been much wider. Therefore, we had to find a way to apply the mathematical model (eq. 3.2) and to get rid of such size dependency of the constituting parameters and transform the dimensional attributes into dimensionless attributes. To achieve it, all the parameters of the equation are divided by a dimensional parameter, which is constant for all the deformed curves obtained from a specific initial curve (see 5.2.2). Knowing the fact that all curves are obtained by applying deformation model over a straight line, it means that the parameter (l, eq. 3.2) remains the same for that set of curves. In order to use the same equation (eq. 3.2) but not to have dimensional parameters in it, a transformation of the equation is obtained as following:

$$NS = \frac{C \cdot A \cdot L}{l^2}\left(\frac{l}{l}\right) = (C \cdot l)\left(\frac{A}{l^2}\right)\left(\frac{L}{l}\right) = C_r \cdot A_r \cdot L_r \quad \ldots\ldots\ldots\ldots\ldots (5.6)$$

Where $C_r = C \cdot l$, $A_r = A/l^2$ and $L_r = L/l$. Using this transformed version of the equation (eq. 3.2) ensures the same values for the measure of straightness, but using dimensionless quantities.

Results:

Based on this reformulation of the measure parameters (5.6), the new values have been computed for the curve initial data set IDS. Afterwards, the same classification algorithm has been applied to all data file curve instances to create the classification model (CMr). In order to evaluate this model, the same set of curves obtained by deformation of initial lines (length of 0.8 and 80 cm) has been used and the results show that this model can correctly classify 99.78 % of all curve instances (Figure 5.11).
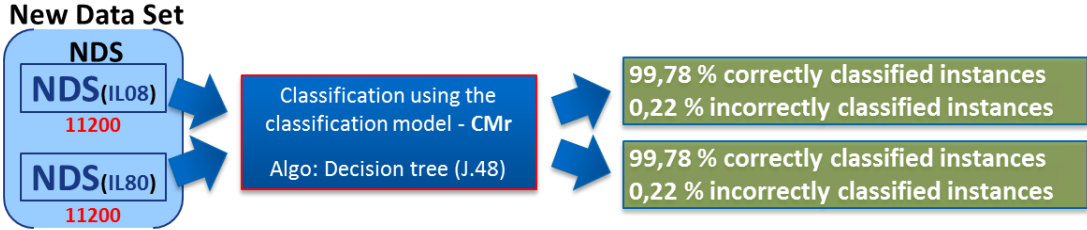

Figure 5.11: Verification of the classification model CMr

Discussion:

As before, the 0.22 % of instances which are incorrectly classified are considered belonging to the neighbor class. Considering that at the border of the class interval (the error is less than 0.5 %), and that the mathematical model (eq. 3.3) for class evaluation is very rigid, a change of 0.5 % in the intervals will not make any visible changes in the appearance of curves. Regarding the validity domain, this classification model can correctly classify 99.78 % of the instances in the interval of 0.8 cm to 80 cm, which is the range of sizes for the most products on the market. Therefore, we can conclude that this classification model is reliable and provides very good classification of curves.

5.6.4 Relevant Attribute selection

Another objective of our experiments was to verify the capabilities of MLTs in identifying the key attributes used for the classification, i.e. those attributes that affect the classification a lot. Thus, we intend to use the attribute selection capability to solve the problem of characterizing free-form curves with respect to their aesthetic properties. Since there exists a specification of styling and aesthetic properties, as well as related measures for curves, the idea was to retrieve them using MLTs. Exploiting such selection capability, we wanted to find out which of the various computable curve characteristics are the most significant for the evaluation and modification of their aesthetic properties. The idea is to adopt the same principle for surfaces for which the most important attributes are not known.

In general, when there is a huge set of data characterized by many different attributes whose importance in the further analysis is not known, Attribute Selection (AS) as a part of Data Mining methodology can be applied. AS allows identifying which attributes can be omitted without affecting the results of further analysis.

To investigate the Attribute Selection capabilities for shape classification, we used the measure of straightness of curves together with its parameters and additional other computable properties for curves. The idea was to see if the Attribute Selection would select the same parameters already used for computation of the straightness measure (S) among a larger list of attributes. In other words, if the Attribute Selection proposes the same curve attributes used in the computation of (S), then we can consider this methodology (AS) as reliable and use it in further investigation of free-form surfaces with respect to the aesthetic properties.

For this purpose, the same set of 11200 curves (IDS(IL8), Figure 5.3) that is part of the initial dataset IDS has been used, and for each curve, the 13 different curve parameters

shown in Table 5.4 are computed. The first three correspond to the relative (transformed) parameters (*Cr*, *Ar*, *Lr*) of the equation (5.6) used for the S measure computation.

Table 5.4: List of geometric quantities used for AS on curves

No.	Parameters	Description
1	L_r	Relative curve length ($L_r = L/l$)
2	A_r	Relative area ($A_r = A/l^2$)
3	C_r	Curvature ($C_r = C * l$)
4	centrox	x - coordinate of barycenter
5	centroy	y - coordinate of barycenter
6	I_x	Moment of inertia over x-axis
7	I_y	Moment of inertia over y-axis
8	Con	The measure of Convexity
9	acceleration	The measure of acceleration
10	acc	Measure of non-acceleration
11	j	Number of local maxima
12	s	The parameter's value of max.
13	ks	Local curvature maximum

Con - corresponds to the measure of convexity which is related to the signed curvature along the curve; acceleration is a measure of acceleration that describes the rising of the curvature along the curve; and acc – is a measure just opposed to the acceleration measure (Giannini, Monti, & Podehl, Styling Properties and Features in Computer Aided Industrial Design, 2004). The other parameters that are closely related to the shape of the curve are the moment of inertia over x-axis (I_x) and over y-axis (I_y). The coordinates (centrox and centroy) of the barycentre can also indicate the shape of the curves. The number of local maximums (j) and its parameter's value(s) can indicate the undulation and wavelike shapes of curves.

Data Mining (DM) task provides two different methods for the AS regarding the attributes evaluation and their representation. The first method uses algorithms that provide independent evaluation of all attributes, and then applies search algorithms to rank all attributes in a list. In this case, the InfoGainAttributeEval – evaluation algorithm is used, which calculates the mutual information (information entropy) of the attributes and classes; then such calculated values are ranked in decreasing order by the Ranker algorithm. The second method uses correlation-based algorithm to evaluate a subset of attributes; then it applies appropriate search algorithms to rank and propose the best subset of attributes. In this case, the CfsSubsetEval – evaluation algorithm is used and then BestFirst search algorithm is applied to propose a subset of attributes that is highly correlated with the classes, but the attributes in the subset are more independent among themselves.

Results:

The following figures show the results of AS in which all attributes associated with the values of the mutual information (first method) are ranked (Figure 5.12), and a subset of attributes is proposed as most significant with respect to the straightness (Figure 5.13).

```
Ranked attributes:
 1.298    1 Relative_Curve_Length(Lr)
 0.896    2 Relative_Area(Ar)
 0.826    7 Iy
 0.793    3 Curvature(Cr)
 0.734    6 Ix
 0.706    5 centroy
 0.437    8 Con
 0.167    4 centrox
 0       12 s
 0       13 ks
 0        9 acceleration
 0       10 acc
 0       11 j
```

Figure 5.12: Ranking of the attributes

```
Selected attributes: 1,2,3 : 3
                     Relative_Curve_Length(Lr)
                     Relative_Area(Ar)
                     Curvature(Cr)
```

Figure 5.13: Selection of a subset of attributes

Discussion:

The results of Figure 5.12 and Figure 5.13 confirm the assumptions previously made: the AS has chosen the same transformed parameters that were used in the computation of the measure of straightness. Comparing the list of ranked attributes and the subset of selected attributes, it can be concluded that the first two selected attributes are ranked as first and second attribute, whereas the third selected attribute appears to be ranked as fourth attribute. Therefore, the AS is very promising for the identification of surface properties meaningful for the evaluation of the aesthetic and styling properties of surfaces and objects.

5.7 Conclusion

The goal of this chapter was to: i) introduce Machine Learning Techniques (MLTs) as a mean for discovering classification patterns with respect to the aesthetic properties of shapes on 2D free-form curves; ii) use Data Mining (DM) methodology to investigate which of the shape characteristics of a geometric entity (here a curve) are the most significant with respect to a specific aesthetic property.

To verify that MLTs could be suitable and useful for shape classification, we have analyzed its behavior in the case of the straightness of 2D curves. We based our work on the mathematical formula for the straightness measure and on the interviews' results achieved in previous works. We verified that MLTs can correctly reapply the classification to new curves. The validity domain of the classifier was tested over a set of curves generated from initial lines that have different dimensions. In addition, we verified the abilities of the Attribute Selection methods to identify the most important attributes among a larger set of attributes. As a result, it was possible to recognize the same curve attributes previously used to compute the measure of straightness (S) as the most characterizing parameters.

Finally, we demonstrated that MLTs are very suitable and can be used efficiently in this kind of engineering applications. This offers good perspectives for solving the same problem on free-form surface, as it will be discussed in the next chapter.

At present, there is no classification of surfaces with appropriate aesthetic properties. This requires, at first, identification of the most meaningful free-form surface characteristics (parameters) and reciprocal relations with respect to the aesthetic properties, and then their classification patterns need to be discovered.

Therefore, the work presented in this chapter is considered to be the first step towards the characterization and classification of free-form surfaces with respect to their aesthetic properties.

Classification of surface shapes

This chapter presents the application of the framework for investigating the existence of a common judgment for the flatness perceived by non-professional designers. Since the perception of flatness of 3D free-form shapes can be affected by the surroundings of the analyzed area, in the analysis, various surroundings have been taken into consideration. Unlike the straightness of curves, the perception of flatness of surfaces is much more complex and requires facing many challenges, as described in Section 6.1. The next section (Section 6.2) presents the framework application and its adaptation regarding the appliance to surfaces. Section 6.3 describes the generation of the dataset of surface instances. This section introduces the diversity of shapes explored (Section 6.3.1) which are later used for creating the Deformation Path and the shape surfaces (Section 6.3.2). By placing the target shape surfaces in different surroundings (Section 6.3.3), the Initial Dataset – IDS is created (Section 6.3.4). Section 6.4 presents the second application of the framework: the geometric quantities related to the surfaces, and the surrounding that will be used for constructing the surface parameters – Attributes. The classification of the IDS by conducting interviews over a group of non-experts is given in Section 6.5. Finally, the experimentations and the results of the framework application are presented in Section 6.6 and in Section 6.7. The last section (6.8) gives a synthesis of this chapter.

6   Classification of surface shapes

6.1   Challenges for surfaces (versus curves)

The surface shapes as geometric entity are a lot more complex than curves, so the analysis of the aesthetic properties of surfaces is faced with many more challenges than for curves. Moreover, there is not available any classification of surfaces or relevant geometric quantities regarding the aesthetic property. As mentioned, the European Project FIORES – II identified terms and initial measures for styling properties of curves, and based on them, (Giannini, Monti, Pelletier, & Pernot, 2013) proposed a refined version of the Straightness measure and of the curve deformation operator for its modification. Differently to curves, for surfaces, no deep analysis of terms and related classification rules has been performed.

Thus, comparing the challenges for the application of the proposed framework in both curves and surfaces, it is evident that for curves, only the base of the temple is missing (creation of the dataset), whereas the other parts of the temple are pretty much well defined. Therefore, we are facing many challenges in order to apply declarative modeling to 3D shapes. While the use of MLTs can provide a valid support for the investigation of classification patterns of the related most relevant geometric quantities of surface, the first challenge to face remains the definition of the appropriate terminology for surface characterization. To this aim, as a starting point in the definition of aesthetic properties of free-form surfaces, Flatness has been taken into consideration as the extension of the straightness for curves. So far, there is neither exact definition of the flatness nor its mapping to the surfaces. We assume that surfaces can be considered flat only if their main sections (or large portions of them) are straight curves. From engineering point of view, a flat surface corresponds to a surface that belongs to a given interval of tolerance defined by two parallel planes required by functional constraints. The distance between those two planes is called the interval of tolerance. From perceptional point of view, a flat surface is not only a plane but also a surface that is "dominantly" flat, where the curvature in both directions does not vary greatly from zero. The curvature is not the only indicator of flatness because there are many shapes that are "dominantly" flat, but they cannot be considered as flat from the perceptional point of view. Same as for curves, where the bounding rectangle indicates the straightness measure, the bounding box of the surface can also be a possible indication of surface flatness. However, it is evident that a direct extension of the curve straightness equation to surface flatness is not possible because there are many other geometric quantities of surfaces that might be strongly correlated to the surface flatness.

Furthermore, for applying MLTs, as for curves, a dataset of surface instances has to be generated. Since the perception of the flatness of a specific surface area depends not only on the area itself but also on the surroundings, the dataset has to include the surface instances given in some surrounding. Next challenge is the identification and computation of geometric quantities related to both the surface area and the surrounding. Such computed geometric quantities have been used in the construction of size independent (dimensionless) surface characteristics. The last and the most challenging task for the application of the proposed framework on surfaces is the classification task. Unlike curves, where the classification of the curve has been carried out by using a formula, there is no such possibility for surfaces. Therefore, having in mind that Data Mining methodology and MLTs require a huge training dataset (few thousands) of instances, we have to find a way how to do it fast, efficiently, and meaningfully. Providing the interviewees with a set of shapes (predefined scenario) for the learning process is more feasible and informative than leaving them to modify the surface shape at their will (user-defined), particularly in our case, in which interviewees may not be familiar with surface modeling tools.

The advantages of predefined scenario are:

- Less time-consuming classification, thus allowing the classification of more surfaces;

- No modeling knowledge is required, thus there is no limitation in the choice of the interviewees;

- Classification of the same shapes by all interview participants, allowing the application of the most commonly used statistical methods for testing the repeatability and stability of the classification.

The drawbacks of predefined scenario are:

- Restricted set of shape characteristic combinations is analyzed because we are defining the target shapes, and the participants are not allowed to introduce or choose among other shapes. Since the participants cannot influence the selection of shapes, the predefined scenario is considered less intuitive.

- Limitation of the participant creativity. This limitation disables the participant to create a shape that might be closer to his/her perception of flatness. This limitation is justified by the fact that introducing too many different shapes will not converge to a solution, since different participants will classify different shapes. Additionally, the wide range of different shapes proposed by the participants will affect the performance of the MLTs in terms of not obtaining reasonable or feasible solution.

The advantages of user-defined scenario are:

- The participant will be able to express examples of what for him represents a flat, or significantly not flat surfaces;

- Allowing the participant to manipulate the shape and modify it makes the classification process more creative and relevant, providing a possibly larger variety of shapes.

The drawbacks of user-defined scenario are:

- User-defined modification scenario is very time-consuming operation requires a powerful computer. This time-consuming operation will either force us to reduce the number of examination shapes (which is not good for data mining methodology), or will dramatically increase the time needed for the classification process. This will seriously affect the classification relevance and the participants will be annoyed;

- It requires familiarity with CAD tools, thus limiting the interviews only to some category of people, which is not our scope aiming at verifying non-professional users' perception.

Regarding the treating of the data acquired during the classification, methods for solving the problems of multiple label classification have to be proposed. Namely, the entire dataset of surfaces has been classified by all participants in the interview, which means the number of participants class labels has been assigned to each shape instance, creating multiple label classification problems.

After various possible modalities to acquire the classification have been introduced, the identification of each part of the framework will be presented in the following sections. In terms of the application of the framework and interpretation of results, the entire dataset of instances has been organized in such a way that allows us to emphasize the different aspects which also influences on the perception of flatness.

The first and most important aspects of this work are to analyze:

- The perception of flatness of every participant in the interviews to investigate if there is common judgment for the flatness from non-professional designers;

- The influence of the context (i.e. surrounding surfaces and type of object) on the perception of flatness.

6.2   Framework application

This chapter describes the application of the framework proposed on surfaces by presenting how each element of the temple is customized to the given application task. The first fundamental element of the framework is the base of the temple that consists of generating and structuring the initial dataset (IDS). It includes the specification and creation of meaningful surfaces and of significant contexts, i.e. type of object and surrounding shapes to be classified. The creation of the IDS is done by performing continuous deformation over three single patches of three different objects: a coffee machine, a car and a car door, and during the deformation, the surface shapes change in five different target shapes. The definition of the context is done by selection of the objects to which the surface belongs, and of the considered extension of the surrounding surfaces. In particular, we considered three types of objects: a coffee machine, a car, and a car door. In order to investigate the influence of the context to the perception of flatness, the context is divided with three different sizes of the surrounding: without context, smaller context, and greater context. If we analyze the market for appliance products, we can conclude that the coffee machines are types of products that can assume a large variety of different shapes. One of the reasons is that the functional constraints of this product are easy to accomplish, so the aesthetic appearance of the product has become a very important factor in its commercial success. Therefore, for faster reach of the customer, the designers try to design pleasant shape of the coffee machines. Another industry where the product shape plays a key role is the automotive industry. The car de-

signers make great efforts in designing the shape of a car that will strongly affect the customers, and will evoke certain positive emotions which will make them buy the product. The current styling and designing of new product relies on the experience and knowledge of the stylists and designers, on which shape they consider would be accepted on the market (i.e. the stylists represent the customers in industrial design process). In other words, the stylists extract knowledge from the customers regarding how they perceive a given shape. Instead of being dependent on the stylist knowledge, in this thesis, the proposed framework makes use of Artificial Intelligence (MLTs) aiming at extracting knowledge directly from the customers. Both the appliance and automotive industry are production fields where the aesthetic appearance of the product is very important, and therefore we decided to adopt product models from these fields.

The second element of the framework refers to the Geometric Quantities of the Entity (surface). In total, we have defined 26 different geometric quantities regarding both the target shape (20) and the context (6), that later have been used in the construction of 36 size independent (dimensionless) surface parameters. Since the flatness can be seen as the extension of the curve straightness in 3D space, it is reasonable to consider that the geometric quantities of curves can be also meaningful for flatness, but extended in 3D space. Therefore, the starting point for constructing the surface parameters (the curves quantities), have been considered.

The third element of the framework is the second column of the temple that represents the part Classification of all instances (surfaces) in IDS. Since there is no classification, for responding to this challenge, interviews over a group of people have been conducted, requesting them to classify all surfaces of the IDS. During the interviews, the participants were asked to classify 8550 surfaces in four different classes (Flat, Almost Flat, Not Flat and Very Not Flat). In order to allow the classification process, a GUI (Graphical User Interface) in Matlab has been created, which is a very intuitive and an easy way to classify surfaces only by moving a slider and clicking mouse buttons.

The fourth element of the framework is the beam of the temple that represents the methods for pre-processing of the acquired data and application of the adopted MLTs. The methods for pre-processing of data (referential and mutual comparison) transform the multi-labeled classification problem into a single-label classification problem, and verify the existence of a common judgment for flatness. In order to respond to the challenges for investigating the influence of the context and different surroundings to the perception of flatness, the initial dataset (IDS) is divided (grouped) in three subgroups, corresponding to different objects and size of the surrounding respectively. The classification of the IDS of 8550 surfaces by each of the participants is saved in a separate ARFF file, obtaining the same number of single-labeled datasets as the number of participants, making it suitable for applying Attribute Selection techniques. Thus After obtaining single-labeled datasets, the first method (CfsSubsetEval + BestFirst) has been applied to all datasets, extracting subset of relevant

attributes for each participant. Surely, by counting the number of times that one parameter is selected from different participants, there will be a group of parameters that will appear more often than others. The attributes selected by the majority of the participants are considered as most relevant parameters of the surfaces, with respect to the flatness.

6.3   Generation of the instances data set

The input to Machine Learning Techniques is a set of instances. As mentioned, each instance is an individual and independent example of the concept to be learned, and it is characterized by the values of sets of predefined attributes. As for curves, it is very easy to imagine a large variety of shapes of a single patch in a product, but many are meaningless or almost impossible to be produced due to the manufacturing capabilities or functional limitations. Therefore, the space of shapes is created in such a way that it contains a shape that, in our opinion, can appear in a real product and at the same time, is meaningful in relation to the examination of the flatness. As explained before, the acquisition of information about the perception of flatness is done by conducting interviews to a group of people. When interviewing people aiming at extracting some knowledge, it is very important to address the following problems from the learning process: 1) selection of the right shapes for the interviews (defining the space of shapes and IDS) and 2) identifying the right order for presenting the shapes. Additionally, by conducting the interviews, our intention is to investigate whether there is a common qualitative judgment for the flatness property for the declarative modeling. Therefore, it can be reasonable to present the slight shape modification in sequences forming a Path. In addition, a repetition of same shape ordering (path) has to be taken into account. The repetition of the same path, in the beginning of each subset, is relevant only in cases when the classification is done by conducting interviews, where the repetition is to address the 'learning stage' of the participant. The learning stage of the participant is necessary for the participants to understand how to classify, and this classification should not be taken into consideration when summarizing the results. The initial dataset (IDS) is created based on a previously defined space of target shapes and by taking a number of intermediate surfaces obtained using Morphing between two target shapes, so that the number of surfaces satisfies the one required by the Data Mining methodology, i.e. up to few thousands of surfaces. Unlike curves, the flatness of surface shapes is investigated also from the point of what is the influence of the surrounding and different objects to the perception of flatness, making them additional objectives in the thesis.

6.3.1 Diversity of shapes explored

The generation of the IDS is based on performing a continuous deformation of single patches of the three chosen objects where during the deformation, each of the surface shape changes in five different target shapes Ts (Figure 6.1). The space of shapes is a set of target shapes that are used to investigate the perception of a given aesthetic property (i.e. flatness).

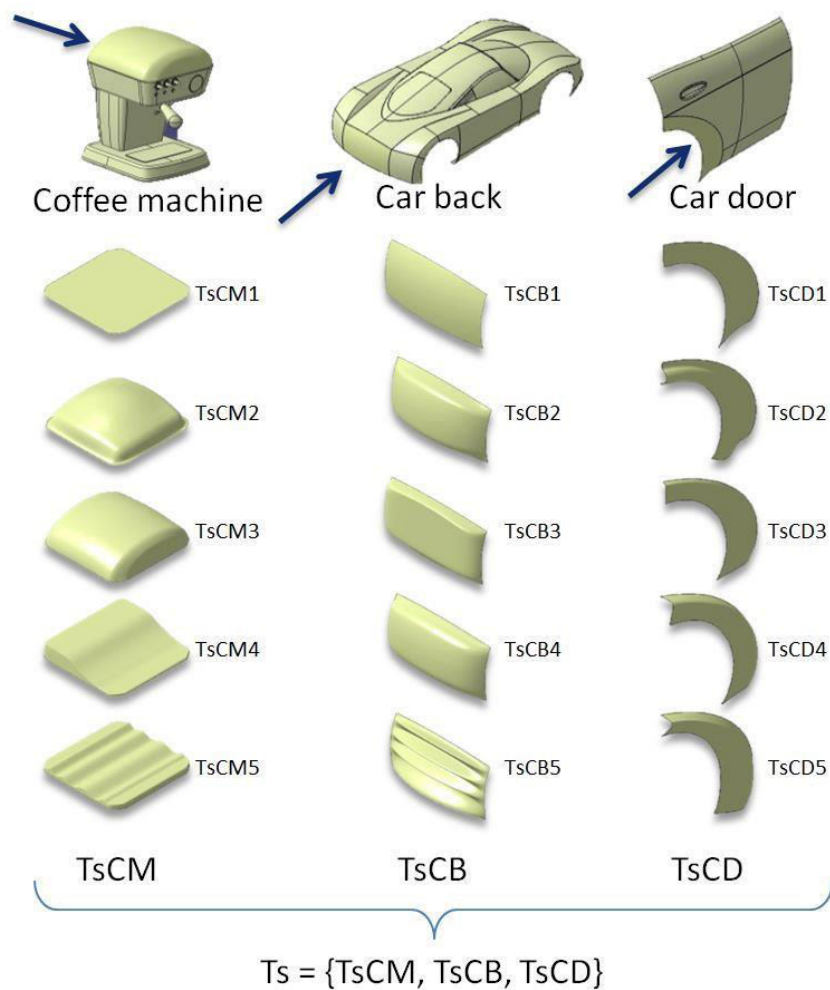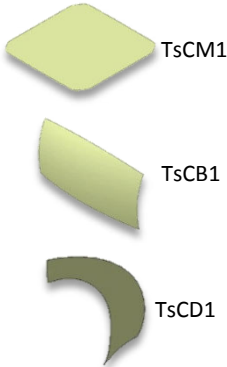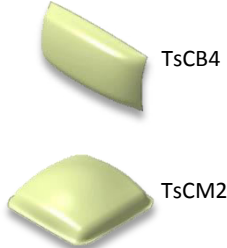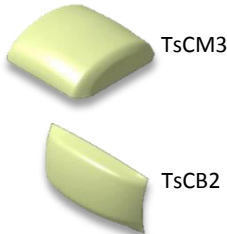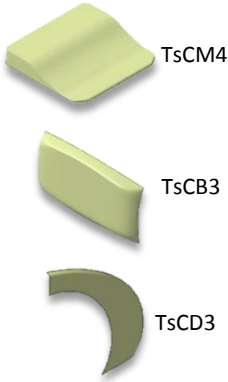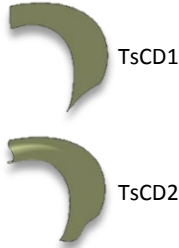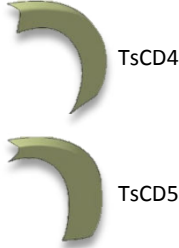**Figure 6.1: The target surface shapes of different objects**
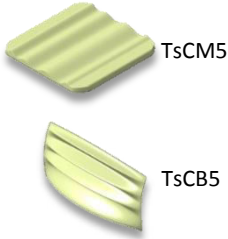
Regarding the manner in which the target surfaces have been chosen, two general principles have been applied:

1. The target surfaces have to be meaningful and can be found in real objects;

2. They have to have as many possible different geometric properties and features.

The reason why the target surfaces have to be of a real achievable shape is because it is insignificant and meaningless to analyze aesthetic properties of shape that would never appear in a real object due to the production or functional constraints. The second principle refers to the need of making a direct relationship between geometric properties and aesthetics, in order to determine the influence of the geometric properties such as symmetry (rotational, one or two planes), asymmetry or the undulation to the perception of flatness. It has to be noticed that not all target shapes are suitable to analyze the same property. For instance, the target shapes of the coffee machine and car back can be considered as more 'regular' shapes than the one of the car door. Hence, the former shapes can be easily used to represent symmetric shapes, whereas the latter shapes are not that convenient to do the same. This is due to the technological and functional constraints of the shapes.

Table 6.1: Target Shapes

The shapes TsCM1, TsCB1 and TsCD1 are the initial surface patches of the three objects and they exist in a real product, whereas the other target shapes are additionally designed in order to represent shapes we want to investigate. The shape TsCM1 has been used to investigate in what way a surface shape with zero curvature can be classified. The difference between the TsCM1 and TsCD1 it that the TsCM1 belongs to one of the origin planes, while TsCD1 belongs to a plane under angle regarding the origin planes (XOY, XOZ or YOZ).	TsCM1 TsCB1 TsCD1
TsCM2 has a bumped shape that has two planes of symmetry. Similar to TsCM2 is TsCB4 which is also bumped with two plane symmetry, but adjusted to the rectangular shape of the initial shape TsCB1.	TsCB4 TsCM2
Rotational symmetry surface occurs when the surface is designed by rotation of a profile curve around an axis of rotation by an angle (from 0° to 360 °). Thus, TsCM3 and TsCB2 are shapes with rotational symmetry, which means the curvature along one direction is always constant, whereas the curvature along the other direction varies depending on the shape of the profile curve (null if the shape is cylindrical).	TsCM3 TsCB2

Another aspect that we were interested in is what the perception of the flatness is when the shape is 'dominantly' flattened, but still occupies some volumes (different from TsCM1). Dominantly flattened means the Mean curvature is zero for more than 80 (85)% of the surface area (TsCM4, TsCB3 and TsCD3) and the rest is a transition, which makes it different from TsCM1.	TsCM4 TsCB3 TsCD3
The shape TsCD2 represents a partial modification of the TsCD1 where one part (almost a half of the TsCD1 surface area) of the surface shape is modified, while the other remains unmodified. This type of modification of the shape aims at investigating how the partial modification of the shape affects the perception of flatness. Not only the portion of modified surface, but also the magnitude of the modification affects the perception of the flatness.	TsCD1 TsCD2
Next, TsCD4 and TsCD5 are shapes with 'sharpened' global modification. Sharpened modifications are types of modifications that result in an appearance of a sharp edge (section with high curvature variation in one direction) in the surface and increasing the overall size (bounding box) of the shape much more than the shape area. Having this type of sharpened shape modification will indicate a possible correlation of the overall shape size (bounding box) with the flatness. TsCM2, TsCM4, TsCB2 and TsCB4 can be considered to have sharpened modification.	TsCD4 TsCD5
We wanted to investigate on what is the influence of the small and 'local' surface modification to the perception of flatness but the overall shape has not changed 'a lot'. Hence, the surface shape TsCM5 and TsCB5 have been designed. The motivation for creating undulated shapes is to test whether the flatness is an aesthetic property of the overall shape or it takes the 'local' variation of the shape into account. In particular, the undulated shapes have been used also for verifying whether the changes in curvature sign affects the perception of flatness.	TsCM5 TsCB5

6.3.2 Definition of the deformation paths and of the morphing process to generate shape surfaces

Each step of modification of the initial surface is performed in such manner that during the modification, it follows a deformation **Path**. A path of deformation is a sequence of target surfaces (TS) for the morphing and is a way of changing the surface shape, aiming at developing another shape that will have different properties and class. The objective of each deformation path is to obtain a wide range of possible shapes that change their geometric properties as much as possible, in order to understand how they affect the perception of flatness. The idea is to have shapes that have different properties, but belong to the same class of flatness, or similar properties but different class. The aim is to understand how a shape can be modified within the same class, and how it can be modified to switch the class. One path can be composed of few or all target surfaces ordered in different sequence. The final **Deformation Path (DP)** of ordered shapes is the collection of all paths together. The Path 1 (for TsCM space of shapes) is composed of the target shapes TsCM1, TsCM2, TsCM3 and again TsCM1 (Figure 6.2). As a first target shape in the path 1, the TsCM1 has been taken because it is a flat surface, and the path 1 finishes again with the same shape (TsCM1). This path provides a possibility for exploring the maximum overall size of the shape, starting from TsCM1, and then reaching the shape with maximum overall size (TsCM3), passing through TsCM2 and returning to TsCM1.
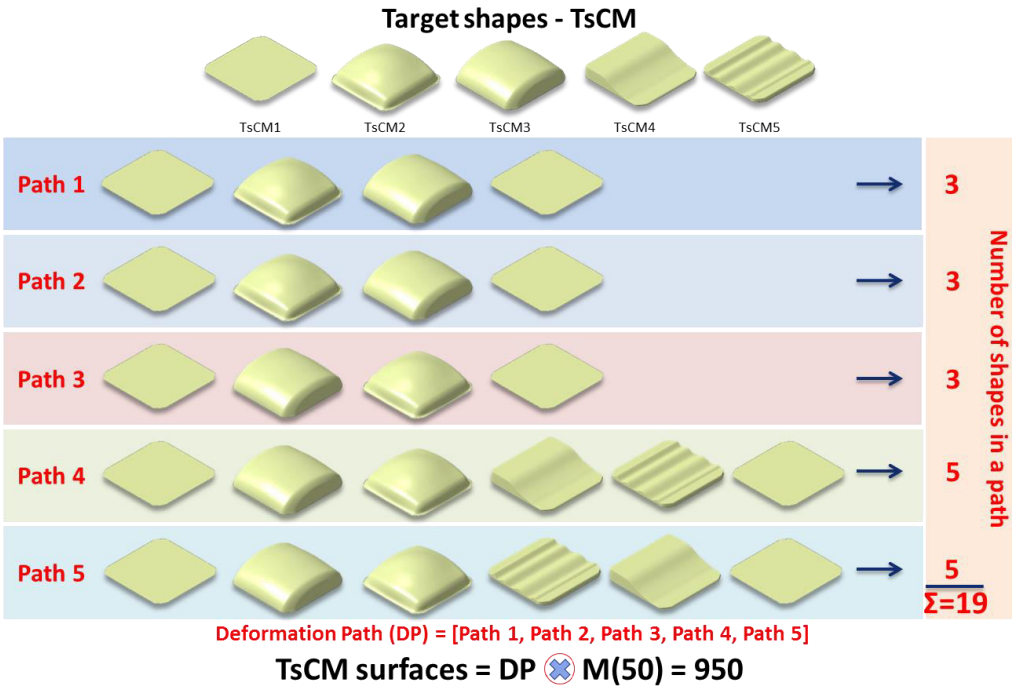

Figure 6.2: Intermediate surfaces used for automatically computing the IDS for the consumer appliance (i.e. Coffee Machine) context

Furthermore, a repetition of same shape ordering (path) has to be taken into account.

The repetition of the same path, in the beginning of each subset, is relevant in our case to address the 'learning stage' of the interview participants. Namely, Path 2 represents a repetition of Path 1 aiming at helping the participants to learn about the classification strategy and improve their classification consistency. After completing the interview with all of the participants, the Path 1 is removed from the dataset IDS, before applying MLTs. Path 3 contains the same intermediate target shapes as Path 1, but differs in the exchanged positions of TsCM2 and TsCM3. The motivation for the position replacement of two shapes is the investigation of the perception consistency and how the shape order affects the perception. Path 4, in fact, is an extension of the Path 3 by adding the remaining two target shapes in between the ones in the third and fourth order position in Path 3. The idea of designing this ordering path is to 'perturb' the interviewees during the classification process by adding other shapes in order to test their classification rules on new set of surfaces. Finally, the last path of shape ordering (Path 5) follows the same idea of Path 4, with a difference in the way that the two additional shapes are shifting their positions. Namely, in Path 5, the shapes TsCM4 and TsCM5 are inverting their order position, so TsCM5 comes before TsCM4 and the path ends again with the shape TsCM1.

The final Deformation Path (DP) for the coffee machine is an accumulation of all 5 paths (Path 1 + Path 2 + Path 3 + Path 4 + Pat 5) of shape ordering connected with TsCM1 shape (because each path starts and ends with this shape). The Collection of all paths of shapes in one set results in creation of the final DP of target shape containing 19 shapes (Figure 6.2). Having in mind that for applying Data Mining we need more than a few thousands of surfaces instances, intermediate surfaces are created applying Morphing (M) operation to change smoothly from one target surface to another at small gradual steps. In our application, **50** intermediary surfaces – M(50) are created between two target surfaces by interpolating of the control points. By applying the Morphing – M(50) to all shapes in the Deformation Path, the set of surface IDS is created. For the coffee machine containing (DP(19) x M(50), Figure 6.2) the same Paths of shape ordering has been adopted for the other two spaces of shapes (TsCB and TsCD), and the corresponding **IDS** have been generated.

The application of the concept of space of shape and set of surfaces provides the possibility of having guided deformation of the shape (predefined scenario of deformation).

6.3.3 Definition of the surrounding surfaces

Beside the investigation of the existence of common judgment for the flatness, the designers also state that the surrounding of a shape has influence on the perception of the shape. Any time when one looks at the surface shape and wants to evaluate the level of flatness, it is almost impossible to remain focused only on the target surface, without taking

into account the nearest surroundings. Intuitively, every time when one is looking at some shapes areas, trying to judge or describe them, the eye focuses on the surface, but (often subconsciously) it moves the focus towards the nearest surroundings and returns back. This phenomenon confirms the consideration that the perception of flatness for a given area might be affected by the surrounding. Another example is, for instance, when we take a computer mouse in our hand, the perception of the shape differs from the perception if the mouse is placed on a table or other wide plane. In order to investigate the influence of the surrounding to the perception of flatness, people have been asked to classify a set of surfaces both out of surrounding (without context), and with two different surroundings (smaller and greater context in Figure 6.3).

Without context   Smaller context   Greater context

Wc1   Sc1   Gc1

Wc2   Sc2   Gc2

Wc3   Sc3   Gc3

Wc   Sc   Gc

Sc = {Wc, Sc, Gc}
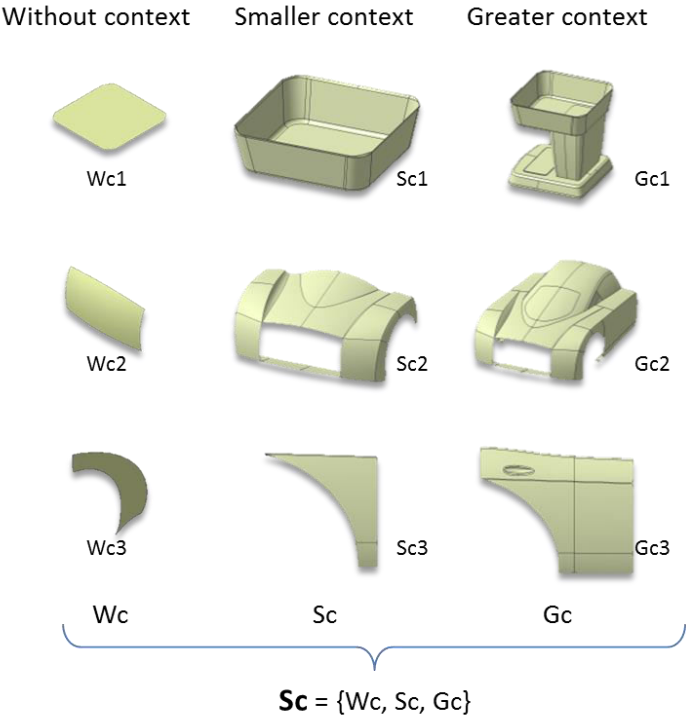
Figure 6.3: Surroundings for the three spaces of shapes

It is true that the perception of flatness is affected by the size of the surrounding, but it has to be independent in some cases. For instance, when someone is classifying sets of surfaces given with different size of surrounding (without, smaller and greater context) in an object, like a coffee machine, he/she follows some classification rules that will not differ greatly from the classification rules he applies to classify the same structure of surfaces, but in other objects (car or car door).

6.3.4 Generation of the initial Dataset of shapes

The initial dataset IDS is created by using Morphing between two target shapes, so that the number of surfaces is expended to the number required by the Data Mining methodology up to few thousands of surfaces.

As explained, the same structuring of the target shapes (Figure 6.2) is applied to the other two target shapes (TsCB and TsCD, Figure 6.1) allowing us create three sets of 950 surfaces, which counts 2850 (3x950) surfaces. Placing the three sets of surfaces (the 2850 surfaces) in three different surroundings (Figure 6.3) respectively, the initial dataset IDS is being created (Figure 6.4) containing 8550 surface instances. Such numerous dataset (8550 instances) is very difficult to be managed during the classification process, so it is necessary to divide it into smaller sets. Therefore, we decided to maintain the division according to the specific paths (Ts) and contexts (Sc), creating (3x3) 9 sets of surfaces. These 9 sets of surfaces are randomly ordered in different sequences for presenting to the participant, regardless of the type of objects and size of the surrounding (Figure 6.5). This ordering of the sets of surfaces is very important in terms of making the participants classify the surface by impression, and not by remembering. For instance, if we present the three sets of surfaces (Wc1, Sc1 and Gc1) of the coffee machine one after the other two, we risk that the participant will classify the first set of surfaces by impression, and the other two by remembering.

Figure 6.4: Initial dataset of surfaces – IDS
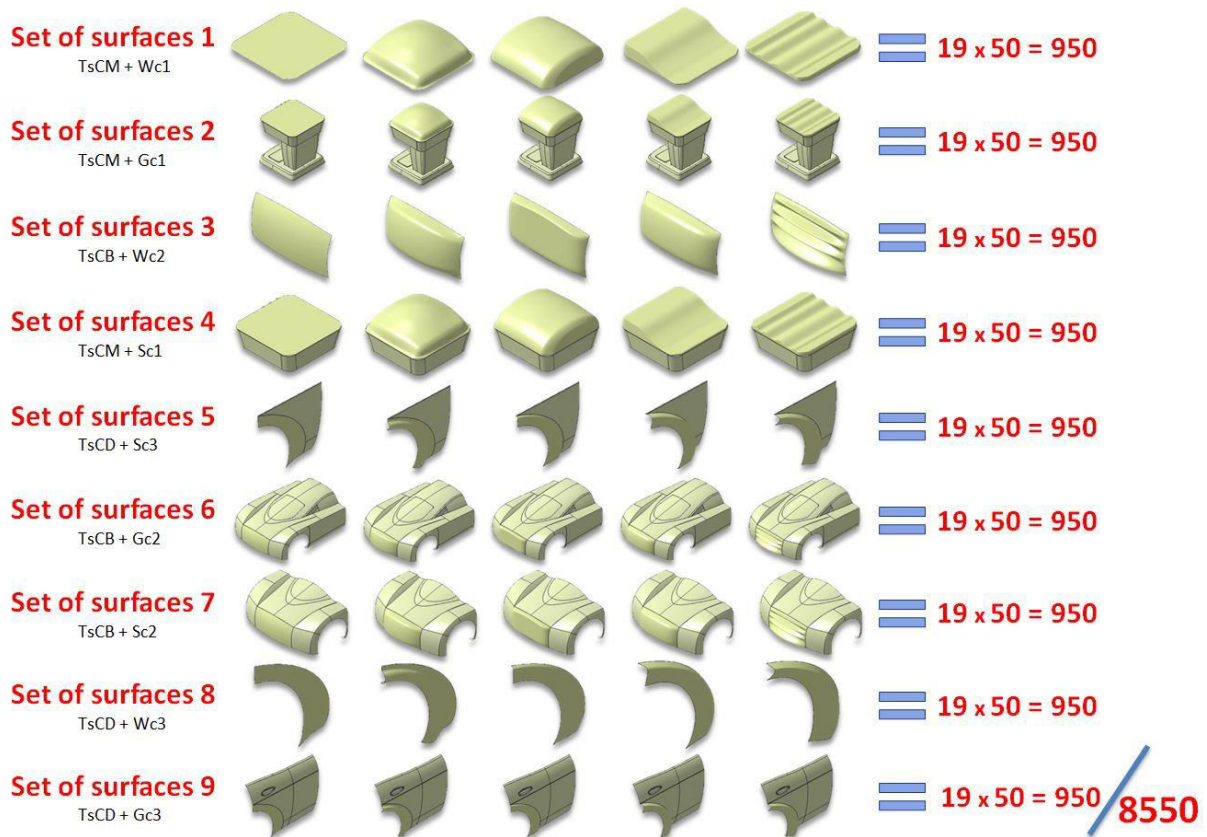
Figure 6.5: Structure of the IDS of shapes used for the interviews

6.4 Definition of surface parameters using basic geometric quantities - Attributes

The perception is a very complex emotional-intuitive activity of the human nervous system where the creation of the mental representation is strongly affected by the past experience, knowledge, cultural and social values. Moreover, the perception is sometimes described as a process of constructing mental representation of the sensory information, shaped by knowledge, memory, expectation and attention. Regarding the classification of shapes by perception, it is common that when people classify shapes in respect to some property, they intuitively follow certain rules and changes of the surface shape. Sometimes, these rules can be explicitly explained, but often, they are implicit and affected by geometric properties of the surface. Aesthetic properties of shapes, as a part of perceptual impression of shapes, are types of properties that are not yet well defined for surfaces and even less mapped to surface properties. Since it has been confirmed, during the interviews, that all participants have followed more or less certain common rules, the objective of this work is to verify if by using MLT methodology we would be able to extract those classification rules, and discover which surface parameters are the most relevant with respect to flatness.

6.4.1 Geometric quantities and surface parameters

Another objective of our experiment is to verify the capabilities of MLTs regarding identifying the most relevant surface parameters with respect to the aesthetic property Flatness. In order to do so, first of all, a set of geometric quantities has been defined together with a mathematical equation for their computation.

The computation of all of these geometric quantities is done in Matlab using a function from IGESToolbox for importing the surface information from IGES (Initial Graphics Exchange Specification) file, and extracting all geometric entities such as: NURBS curves and NURBS surfaces, trimmed patches, points, and surface triangulation. Next, using standard mathematical equations (Heron's formula for computation the area of any triangle) and operations (projection of points cloud onto plane along projection vector, PCA principles for object orientation vectors, first and second derivation in a given point), we are able to compute surface area (A_s) and its projection (A_p), volume of the bounding box (V), normal vector (N_a), Principals Curvatures and so on (Table 6.2).

In Chapter (5) are presented the results of the application of the MLTs for evaluation of the capabilities of learning algorithms, in terms of classification of curves and choosing the most relevant attributes in respect to the Straightness. This evaluation recommends the use of size independent geometric attributes, such that classification models created in this manner do not depend on the overall size of the geometric entity (curves or surfaces), which leads to the highest classification accuracy and better selection of relevant attributes. On the one hand, we are able to compute the geometric quantities for the target surfaces given in Table 6.2 and Table 6.3 which are dimensional attributes, but on the other hand, the use of size independent (dimensionless) attributes is essential for the reasons explained in the previous chapter (Section 5.6.3). Therefore, new size independent surface parameters based on the geometric quantities are defined. The definition of the new surface parameters has to match the criteria of size independence (dimensionless) and in the same time, the choice of the geometric quantities for definition of the parameters have to be relevant in respect to the property. Most common way of obtaining size independent parameters is to construct ratios between two geometric quantities or two groups of geometric quantities of identical nature (dimension). As previously said, the construction of ratios is an easy way of obtaining dimensionless parameters, but in the same time, it suffers from the risk of getting infinite value for the ratio in case the denominator is zero. This can be avoided only if the denominator is not a single geometric parameter, but a summed group of few geometric parameters, ensuring that at least one of these parameters is different from zero. Considering that the flatness can be seen as the extension in 3D space for 3D free-form surfaces of the straightness as aesthetic property of 2D free-form curves, it is natural to conclude that the parameters of the curves, used into the definition of straightness, can be also meaningful in respect to the flatness, after being translated in 3D space.

Unlike the curves, where at a given point there is only one tangent and thus only one value of curvature, the free-form surface at a given point does not have no single value for the curvature. Therefore, the principal curvatures have been considered (k_1 and k_2). These curvatures are the basics for computing the Gaussian (G_c), Mean (M_c) and Absolut (A_c) curvature:

$$G_c = k_1 * k_2$$

$$M_c = (k_1 + k_2)/2$$

$$A_c = |k_1| + |k_2|$$

In the 3D space, the area (A) of the straightness measure can have as a counterpart the volume (V_s) between the surface and a plane (one of the minimum bounding box planes). The length of the curve L in the 3D space will/can be substituted by the surface area (A_s). Lastly, the length l between the curve end points can be seen as the area (A_p) of the region obtained by projecting the surface onto a reference plane. The selection of the reference plane, which will be used as a projection plane for computing the projection area (A_p) and volume (V_s) of the surface, has to be independent from the surface position. The necessity of this constraint lies in the fact that the values of the geometric quantities need to depend only on the shape of the surface and not on its creation process or on its position or orientation. For instance, the reference system planes (XOY, XOZ and YOZ) are not convenient to be used as reference planes since changing the orientation or position (e.g. the surface is rotated or translated) results in a change of the volume (V_s) underlying the surface regarding to any reference planes, whereas the surface shape has not changed at all. In order to have position and orientation independence, the referent plane has to be computed only by using intrinsic information of the surface. The simplest intrinsic information of a surface that in the same time preserves its shape is the 3D point cloud extracted from the surface. There are two ways how a referent plane can be computed: 1. best fitting plane of the 3D point cloud and 2. one of the faces of the minimal Bounding Box.

The first way is by direct approximation of a plane into the surface 3D point cloud. The direct approximation of a plane is rotation and position independent, but not deformation independent. Namely, if the surface changes its shape, even locally, both the referent plane position and the values of the geometric quantities related to it will change without depicting the real changes of the shape. For instance, when we compute the volume between the surface and the reference plane, after the deformation of the surface, the change of the volume has to be affected only by the change of the shape and not affected by the reference plane. If we want to compare two geometric quantities, we have to compare them regarding the same reference. Therefore, the referent plane has to retain the same position after the surface shape is modified. Since we are interested in slight modifications of the surface, it is reasonable to assume that during the modification, the bordering curves (minimum one) and corner points (minimum 2) remain unchanged. This leads to the conclusion that the refer-

ence plane has to be related to those elements. The second way is by applying PCA (Principal Component Analysis) to compute the 3D point cloud orientation vectors (3 eigenvectors). By considering the dot product between position vectors of all points and the 3 eigenvectors, it is possible to find the most distanced points along all 3 eigenvectors (6, 3 pairs of most distant points). The 6 most distant points and the 3 eigenvectors are sufficient information for computing the minimal Bounding Box (MBB). The MBB consists of 6 planes, 3 pairs of parallel planes (A_1, A_2 and A_3). Since we want to investigate the perception of flatness considering also the influence of the surrounding, the deformation of the surface should not affect the boundaries with the surrounding in order to preserve at least the G0 continuity avoiding the gap appearance. Therefore, we consider that during the modification of the surface, the surface bounding curves and corners points remain unchanged. The MBB plane that is nearest to the corner points can be considered as deformation independent plane, and therefore this plane is used as reference plane. Another restriction concerning the modification of the surfaces is that the trimming and cutting operation onto surfaces has not been taken into account. Other geometric quantities related to the MBB are its volume (V), the length of its diagonal (D) and the length of its edges (E_1, E_2 and E_3).

Finally, for selecting the most relevant surface parameters, in respect to the aesthetic property Flatness, in our work, 36 surfaces parameters have been specified using the previously defined set of geometric quantities of surface. The set of geometric quantities is divided in two subsets. The first subset represents the geometric quantities related to the surfaces to be classified (Figure 6.6 and Table 6.2), whereas the second subset of parameters includes those related to the surrounding (Figure 6.7 and Table 6.3).
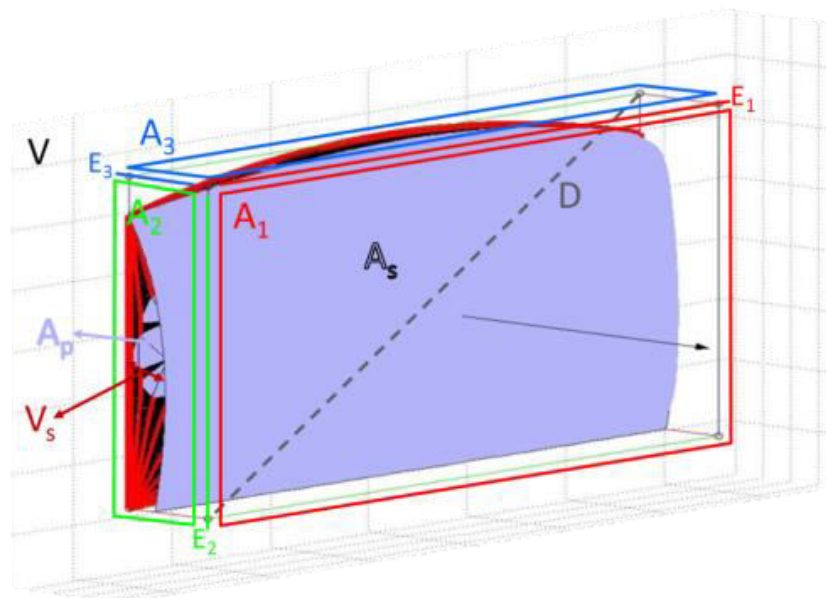
Figure 6.6: Geometric quantities of a surface

Table 6.2: Geometric quantities of a shape

A_s – Surface area

A_p – Area of the projection of the surface on the plane with the smallest normal PCA vector

V_s – Volume that is occupied between the surface and its projection

V – Volume of the minimum bounding box of the surface

A_1 – The area of the biggest face of the minimum bounding box

A_2 – The area of the second biggest face of the minimum bounding box

A_3 – The area of the smallest face of the bounding box

D – Diagonal of the minimum bounding box

E_1 – The longest edge of the minimum bounding box

E_2 – The second longest edge of the minimum bounding box

E_3 – The shortest edge of the minimum bounding box

M_c – Mean curvature ($\frac{1}{p}\sum_1^p H_i$), where p is the number of the surface discretization points and H_i is the mean curvature value on the *i-th* point on the surface

G_c – Gaussian curvature ($\frac{1}{p}\sum_1^p K_i$), where p is the number of the surface discretization points and K_i is the Gaussian curvature value on the *i-th* point on the surface

A_c – Absolute curvature ($\frac{1}{p}\sum_1^p A_i$), where p is the number of the surface discretization points and A_i is the absolute curvature value on the *i-th* point on the surface

N_a – Average normal of the surface ($\frac{1}{p}\sum_1^p N_{ai}$), where p the number of the surface discretization points and N_{ai} is the normal value on the *i-th* point on the surface

R_p – Radius of a sphere that has same area as A_p ($R_p = \sqrt{\frac{A_p}{4\pi}}$)

R_s – Radius of a sphere that has same area as A_s ($R_s = \sqrt{\frac{A_s}{4\pi}}$)

N_p – Percentage of surface with positive Gaussian curvature (%)

N_n – Percentage of surface with negative Gaussian curvature (%)

N_z – Percentage of surface with zero Gaussian curvature (%), $N_p + N_n + N_z = 1$
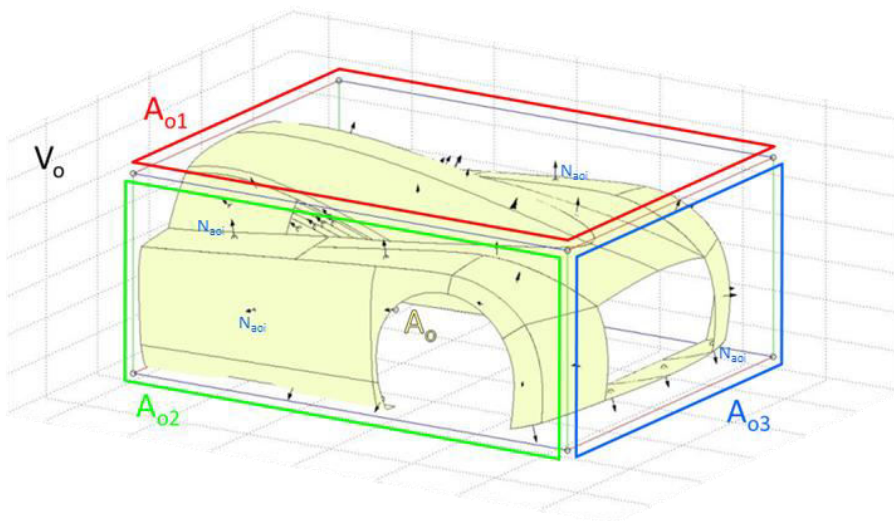
Figure 6.7: Geometric quantities related to the surrounding

V_o – Volume of the bounding box of the object

A_o – Total area of the object

A_{o1} – Area of the biggest plane of the object bounding box

A_{o2} – Area of the second biggest plane of the object bounding box

A_{o3} – Area of the smallest plane of the object bounding box

N_{ao} – Average normal of the surrounding surface patches in the object ($\frac{1}{p}\sum_1^p N_{aoi}$), where p is the number of the surface patches discretization points and N_{aoi} is the normal value on the *i-th* point on the surface

36 surface parameters are defined as ratios of the listed geometric surface quantities. As previously said, the construction of the surface parameters has to satisfy two requirements. The first requirement is related to the selection of geometric quantities that are highly relevant with respect to the flatness. The second requirement is construction of surface parameters that must be dimensionless. The list of the 36 surface parameters is divided in two subsets of surface parameters. The first subset of surface parameters are ratios related to the surface shape that is modified during the deformation, whereas the second subset includes the surrounding-related parameters in which the overall size of the context affects the perception of flatness. In the following, both subsets of surface parameters are listed and detailed:

1. Ratio between the surface area A_s and its projection A_p:

$$R_1 = \frac{A_s}{A_p}$$

Intuitively, the surface area has a great influence on the perception of flatness; same as the curves length (L) to the straightness. In order to have size independent surface parameter, the surface area is divided by its projection area (A_p) over A_1 plane of the MBB. Since during the deformation, the bordering curves and corner points remain the same, the surface projection area stays the same as well, and R_1 becomes a size independent quantity for the surface area. For a flat surface $R_1 = 1$, bloating a flat surface (A_s) increases, whereas the ratios R_1 increases and vice versa.

2. Ratio between the surface volume V_s and bounding box volume V:

$$R_2 = \frac{V_s}{V}$$

The volume V_s of the surface indicates how much space the surface occupies in relation to the reference plane (A_1), while R_2 indicates the overall occupancy of the surface volume inside the minimal Bounding Box. For a given surface and bounding box, the growth in value of R_2 fills more space in MBB, which leads to changes of the surface shape. The drawback of this parameter is that when the surface is flat, the volume is zero, which is the same as the volume of MBB. The ratio of two zeros is undefined and this problem has been solved using mathematical approximation, so that the value R_2 is approximate to zero if the MBB volume is lower than 0.00001.

Other geometric quantities related to the Bounding Box that can also be considered as indicators of their possible correlation to the surface flatness are the so called edges of the MBB. A bounding box has three different edges: E_1, E_2 and E_3, and to normalize their values in terms of proportion, the Diagonal (D) of the MBB is used. In some way, the edges of the MBB represent a 'thickness' of the MBB in the given direction and in the same time a 'thickness' in the three directions of the surface shape. Therefore, the purpose of these parameters is to investigate whether the 'thickness' of the surface shape affects the perception of flatness. The Diagonal D assumes value zero only if the surface becomes a point, which is not possible. I compute $D^2 = E_1^2 + E_2^2 + E_3^2$ so if the surface is planar, it means that the smallest edge is zero (E_3), so the final equation will be in the following form: $D^2 = (E_1^2 + E_2^2)$, which represents the diagonal of one of the faces of MBB. The considered ratios, including the MBB characterizing quantities, are:

3. Ratio between the longest edge E_1 and the diagonal D of the Bounding Box:

$$R_3 = \frac{E_1}{D}$$

4. Ratio between the second longest edge E_2 and the diagonal D of the Bounding Box:

$$R_4 = \frac{E_2}{D}$$

5. Ratio between the smallest edge E_3 and the diagonal D of the Bounding Box:

$$R_5 = \frac{E_3}{D}$$

6. Ratios between the dimensions of the Bounding Box:

$$R_6 = \frac{E_2}{E_1}$$

$$R_7 = \frac{E_3}{E_2}$$

$$R_8 = \frac{E_3}{E_1}$$

The last geometric quantities, related to the bounding box, i.e. the areas of the MBB planes A_1, A_2 and A_3, are also included in the investigation for tracking their possible correlation to the perception of flatness.

Firstly, the ratios (R_9, R_{10} and R_{11}) between the areas of each of the MBB planes are computed, aiming at exploring the interrelation of MBB planes and the mutual influence to the perception of flatness.

7. Ratio between the areas of the planes of the bounding box

$$R_9 = \frac{A_2}{A_1}$$

$$R_{10} = \frac{A_3}{A_2}$$

$$R_{11} = \frac{A_3}{A_1}$$

8. Secondly, the ratios (R_{12}, R_{13} and R_{14}) represent the influence of the MBB plane areas with respect to the area of the bounding box. The goal of tracking these ratios is to examine the possible correlation of all particular MBB plane areas to the perception of flatness.

$$R_{12} = \frac{A_1}{A_1 + A_2 + A_3}$$

$$R_{13} = \frac{A_2}{A_1 + A_2 + A_3}$$

$$R_{14} = \frac{A_3}{A_1 + A_2 + A_3}$$

As explained before, at a point of a free-form surface there is an infinite number of tangent curves and hence curvatures among which the minimum and maximum curvatures are called Principal curvatures. These two principal curvatures are the basics for computing the Gaussian (G_c), Mean (M_c) and Absolut (A_c) curvature. Therefore, the group of surface parameters from R_{15} to R_{24} refers to those geometric quantities

that are additionally multiplied by some other geometric quantities, aiming at preserving their size independent character. Accordingly, the values of the Mean and Absolute curvatures are multiplied with four geometric quantities to transform them into dimensionless surface quantities. Since the unit for Mean and Absolut curvatures is length unit to the minus one (in our case is m^{-1}), the overall unit of the geometric multipliers has to be the same length units (m). The first multiplier (R_p) corresponds to the radius of a sphere having an area equivalent to the projection area (A_p) of the surface (Table 6.2). Since the projection of the bounding curves of a given surface remains unchanged during the deformation of the shape, the Radius (R_p) will be constant, so that the computed surface parameter will reflect only the changes of the curvatures. On the contrary, the second multiplier is the radius of a sphere having area equal to the area of the surface (A_s), which is different for different surface. This second multiplier takes into account the surface area, and, by multiplying the curvature related quantities (M_c and A_c) constructs a surface parameter that incorporates the influence of both the surface area and the curvature simultaneously. Next, the third and fourth multipliers are the ratios between the surface volume (V_s) and the projection area (A_p) or surface area (A_s), respectively. Unlike the units for Mean and Absolute curvature, the unit for the Gaussian curvature is length units to the minus two (m^{-2}), which means that the multiplier has to have the same length unit square (m^2) to transform the Gaussian curvature into a dimensionless surface parameter. Similar to the previously mentioned multiplier, for computing the multipliers, the surface area (A_s) and its projection (A_p) have been used. Thus, the surface properties R_{19} and R_{20} are computed multiplying the Gaussian curvature by surface projection area (A_p) and the surface area (A_s) respectively.

9. Multiplication of the Mean curvature with the Radius R_p of a sphere that has the same area as the surface projection area (A_p)

$$R_{15} = M_c * R_p$$

10. Multiplication of the Mean curvature with the Radius R_s of a sphere that has the same area as the surface area (A_s)

$$R_{16} = M_c * R_s$$

11. Multiplication of the Mean curvature with the ratio between the surface volume V_s and the surface projection area A_p

$$R_{17} = M_c \frac{V_s}{A_p}$$

12. Multiplication of the Mean curvature with the ratio between the surface volume V_s and the surface area A_s

$$R_{18} = M_c \, \frac{V_s}{A_s}$$

13. Multiplication of the Gaussian curvature and the surface projection area A_p

$$R_{19} = G_c * A_p$$

14. Multiplication of the Gaussian curvature and the surface area A_s

$$R_{20} = G_c * A_s$$

15. Multiplication of the Absolute curvature with the Radius R_p of a sphere that has the same area as the surface projection area A_p

$$R_{21} = A_c * R_p$$

16. Multiplication of the Absolute curvature with the Radius R_s of a sphere that has the same area as the surface area A_s

$$R_{22} = A_c * R_s$$

17. Multiplication of the Absolute curvature with the ratio between the surface volume V_s and the surface projection area A_p

$$R_{23} = A_c \, \frac{V_s}{A_p}$$

18. Multiplication of the Absolute curvature with the ratio between the surface volume V_s and the surface area A_s

$$R_{24} = A_c \, \frac{V_s}{A_s}$$

The last group of surface parameters includes the parameters R_{25}, R_{26} and $R_{27,}$ providing the information of the percentage of the surface with positive/negative/null Gaussian curvature. The average values of the surface curvature (G_c, M_c and A_c), mentioned earlier, are too rough to indicate the shape of the surface. Therefore, in order to have a better idea of the surface shape, a group of surface parameters is defined so as to indicate the distribution of the curvature (Gaussian) on the surface. So, if N − is the number of points on the surface for which the Gaussian curvature is computed, and N_p N_n and N_z are the number of points that have positive, negative and zero curvature respectively, then the ratios between N_p, N_n and N_z and the total number of points give the surface properties R_{25}, R_{26} and R_{27} respectively.

19. Positive curvature

$$R_{25} = \frac{N_p}{N}$$

20. Negative curvature

$$R_{26} = \frac{N_n}{N}$$

21. Zero curvature

$$R_{27} = \frac{N_z}{N}$$

The average normal N_a is a geometric quantity (Table 6.2) computed for each surface. First, the normal vectors (N_{ai}) at all surface discretization points (p) have been computed. The components (X, Y and Z) and the average normal vector N_a are computed by averaging the corresponding components (X_{ai}, Y_{ai}, and Z_{ai}) for all points. Thus computed, average normal vector represents the global orientation of the surface, and its length accumulates the distribution of all single normal vectors (N_{ai}). The distribution of the normal at the points depends on the surface shape. For instance, if the surface has smooth changes of the shape without local modification (bumps and hollows), this leads to the conclusion that all the normal vectors (N_{ai}) have approximately the same tendency of their directions. When a group of vectors has approximately same directions, then the average vector has bigger length than the one of a group of vectors with random directions. Since the average normal depends of the surface shape, we want to investigate whether there is a strong correlation between surface normal and the perception of flatness.

22. Average normal $|N_a|$

$$R_{28} = |N_a| = \sqrt{X^2 + Y^2 + Z^2}$$

$$X = \frac{1}{p}\sum_1^p X_{ai}, \;\; (Y = \frac{1}{p}\sum_1^p Y_{ai}), \;\; (Z = \frac{1}{p}\sum_1^p Z_{ai}) \text{ and } N_{ai} = (X_{ai}, \; Y_{ai}, \; Z_{ai})$$

The second subset of parameters is conceived to check if the size of the context affects the perception of flatness. Therefore, as a representative of the surrounding size, without considering its shape, is the object Bounding Box. Thus, the tracking of the surrounding size influences to the perception of flatness being reduced to comparing the sizes of the portion of the object including the surrounding (in the following indicated as object for simplicity)

and analyzed surfaces Bounding Boxes. It was explained before that the minimal Bounding Box of the surfaces has been used for constructing some of the surface parameters, whereas when considering the object, it is not known whether the minimal Bounding Box is suitable or not. In our opinion, from the perception point of view, the bounding box of the object has to be positioned so that all of the planes are parallel to the ones of the surface bounding box. Having in mind that both the position of the surface within the surrounding and the surrounding shape are arbitrary, it is not guaranteed that the minimal Bounding Box of the object is 'parallel' to the one of the surfaces. Therefore, the bounding box of the object has been constructed using the same orientation vectors (eigenvectors) of the surfaces to ensure that both bounding boxes have the same spatial orientation. In the following, the second subset of surface parameters is detailed while explaining the rationale for their choice. To distinguish among the quantities related to the surface and those associated to the surrounding, these last ones have the suffix 'o'.

23. Ratio between the surface area and the area of the objects

$$R_{29} = \frac{A_s}{A_o + A_s}$$

The first of second subsets of parameters is the ratio (R_{29}) between the surface area (A_s) and the total area of the object (A_o). The reason why the denominator comprises both the surface area (A_s) and the object area (A_o) is the need to be able to compute this parameter even when only the surface is considered without the surrounding (context). When the surface is out of context, the object area (A_o) is equal to zero and the value of the ratio (R_{29}) will be 1 for all the surfaces that are out of context. Instead of adding the surface area (A_s) in the denominator, a randomly chosen number, e.g. 1, can be added. The use of a constant number in the denominator will make the value of the ratio (R_{29}) strongly correlated to the surface area (A_s) when the surfaces are out of context. In fact, this ratio has to depict the influence of the surrounding to the perception of flatness, so if there is no surrounding then R_{29} should remain unchanged, which is not the case if a constant number has been used instead of A_s. Finally, the goal of this surface parameter is to investigate whether the total surface area of the object (A_o), as an indicator of the surrounding size, affects the perception of flatness for the target surfaces.

24. Ratio between surface and object bounding box volumes

$$R_{30} = \frac{V}{V_o + V}$$

Another parameter that may indicate the possible influence of the surrounding shape size to the perception of flatness is obtained by relating the volumes of the surface minimal Bounding Box (V) and of the Bounding Box of the object (V_o). The bounding box of the object is not the minimal BB, but the BB created using the eigenvectors of target surfaces, for the reason explained earlier.

25. Ratio between surface volume and object bounding box volume

$$R_{31} = \frac{V_s}{V_o + V_s}$$

The second volume based parameter is constructed by computing the ratio between the surface volume V_s (Table 6.2) and the object volume V_o. Similar to the previous surface parameter, the ratio R_{31} is an alternative way of tracking the possible influence of the surrounding size to the perception of flatness.

26. Ratio between the smallest plane of the surface MBB (A_3) and the plane of the object BB parallel to it (A_{o1})

$$R_{32} = \frac{A_3}{A_{o1} + A_3}$$

27. Ratio between the second biggest plane of the surface MBB (A_2) and the plane of the object BB parallel to it (A_{o2})

$$R_{33} = \frac{A_2}{A_{o2} + A_2}$$

28. Ratio between the biggest plane of the surface MBB (A_1) and the plane of the object MBB parallel to it (A_{o3})

$$R_{34} = \frac{A_1}{A_{o3} + A_1}$$

Other than the volume of the BB, the areas of the BB planes have also been used in the construction of surface parameters. All these parameters are used to investigate whether the BB planes of the object (A_{o1}, A_{o2} and A_{o3}), as an indicator of the surrounding size, affect the perception of flatness for the target surfaces.

29. Ratio between the diagonal of the surface bounding box D and the diagonal of object bounding box

$$R_{35} = \frac{D}{D_o + D}$$

The last surface parameter related to the bounding box is the ratio R_{35} between the space diagonal of the target surfaces MBB (D) and the space diagonal of the object BB (D_o). As for the previous surface parameter, the purpose of the R_{35} is to investigate if the ratio between space diagonals of the appropriate Bounding Boxes is highly correlated to the perception of flatness.

30. Distribution of the normal

$$R_{36} = \frac{1}{k} \sum_1^k \frac{dot(N_a, N_{aoi})}{|N_a| \, |N_{aoi}|},$$

In addition to the comparison of the bounding boxes of both the object and target surface, the comparison of the average normal vector to the normal vectors of the surrounding can also indicate a possible influence of the surroundings to the perception of flatness. In fact, in our opinion, the shape of the surrounding surface patches influences the perception of flatness. Namely, the perception of flatness is not the same when the surface is surrounded by surfaces that are, more or less, 'coplanar' (car door, Figure 6.1), or when there is a strong change in the curvature in the surrounding connection, e.g. the surface is surrounded by surfaces that are positioned under an angle greater than 90° (coffee machine, Figure 6.1). Therefore, shape changes taking into consideration the relationships between target surface and surrounding surface patches, the dot product between the average normal vector (N_a) of the surface and all (k) surrounding surfaces (N_{aoi}) is divided by the vector norms of both the considered and surrounding surfaces average normal vector. The R_{36} is average of the cosine of the angle between the average normal of the target surface and k average normal of the surrounding surfaces.

It is very important to emphasize that this list of surface parameters does not exhaust the entire set of all possible surface parameters. The geometric quantities used to compute these parameters are extending the geometric quantities of curves in the 3D space for surfaces. In addition, the aspect of surrounding and its influence over the perception of flatness, which is not treated for curves, is a complementary part that enriches the understanding of the perception of aesthetic properties of surfaces. This list of surface properties is an attempt to approach to as much as possible relevant surface parameters with respect to the flatness. Of course, the list of surface parameters is not limited only on those parameters listed above, for there are many others, but they are mostly not relevant with respect to the flatness. Many of these parameters can be constructed using different reference planes that do not preserve their position when the surface changes its shape.

6.5   Classification of the surfaces by carrying out interviews

The third element of the framework and the second column of the temple is the classification of all instances (surfaces) of the IDS. Before starting with the classification process, the classification classes of flatness need to be defined. The number of classes has to be, in some way, a compromise between a very fine classification aimed at extracting as much information as possible from the interviewees, and a very limited one, to not confuse the participants. In other words, if we propose two classifications and we classify the IDS in two classes (for instance, flat and not flat), we will not be able to extract any relevant and significant information or classification patterns. Using two classifications would provide very poor and irrelevant knowledge because only a very limited set of the instances would be classified as flat and the remaining part would be classified as not flat (depending on the IDS). The other extreme situation would be if we proposed 5 or more classifications; the participants would be confused and would be faced with ambiguity in distinguishing the difference between some classes. Having too many classes would impede the classification process and would affect the quality of the acquired information. For instance, if we asked a person to classify 8,550 surfaces in 5 or more classes, in any given moment, the participants would be in a dilemma whether to classify the shape into the fourth or the fifth class. This ambiguity would affect the classification process in terms of time consumption and would extend the time needed for the classification Therefore, in this research, we decided to propose four classifications that we believe match the criteria for the optimal number of classes (not too few and not too many). Figure 6.8 below shows the four classes of flatness that have been defined for the classification process.
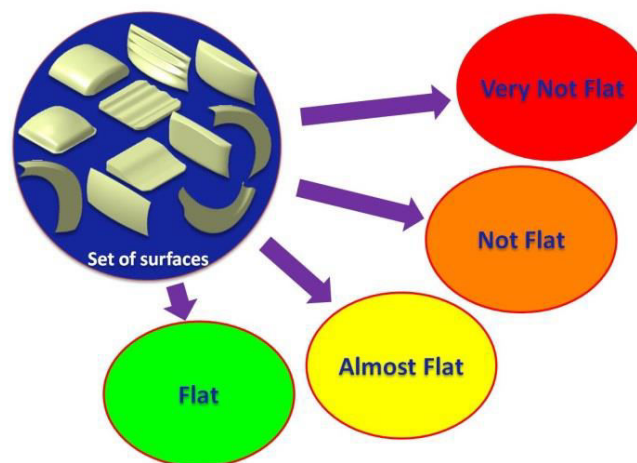
Figure 6.8: Classes of flatness

The next important aspect of defining the classification classes is also defining the vocabulary for each of the classes. The naming of the classes has to be more intuitive and close to the layperson's perceptions, avoiding the use of geometrical or mathematical terms. Thus, we opt for quantitative judgments (e.g., less, almost, more, very, not, or not at all) of these

classes of flatness. Therefore, we propose four levels of classification of flatness: Flat, Almost Flat, Not Flat, and Very Not Flat. A Flat surface in mathematics is a surface that has zero curvature in all points along both (u and v) directions, and not only the Gaussian but also the Mean curvature is zero (a plane), but from a perception point of view, a flat surface can include a range of surface with very small bumping. An Almost Flat surface is a surface that tends to be flat but is still not flat (small bumping). A Not Flat surface is a surface that deviates much more from a flat (or plane) surface, and it represents a kind of bumped surface. The last class, a Very Not Flat surface, is a surface that deviates too much from a flat (plane) surface.

As previously said, one of the most important requirements for applying DM methodology is having a few thousands of instances. If we want to satisfy this demand and request from each participant to classify a few thousands of surfaces printed on a sheet of paper, the classification will take more than a few hours, which is neither acceptable nor feasible. In order to expedite the classification process, a GUI (Graphical User Interface) in Matlab has been created (Figure 6.9) that allows classification of surfaces in the four classes in a very intuitive and easy way, only by moving a slider and clicking buttons.
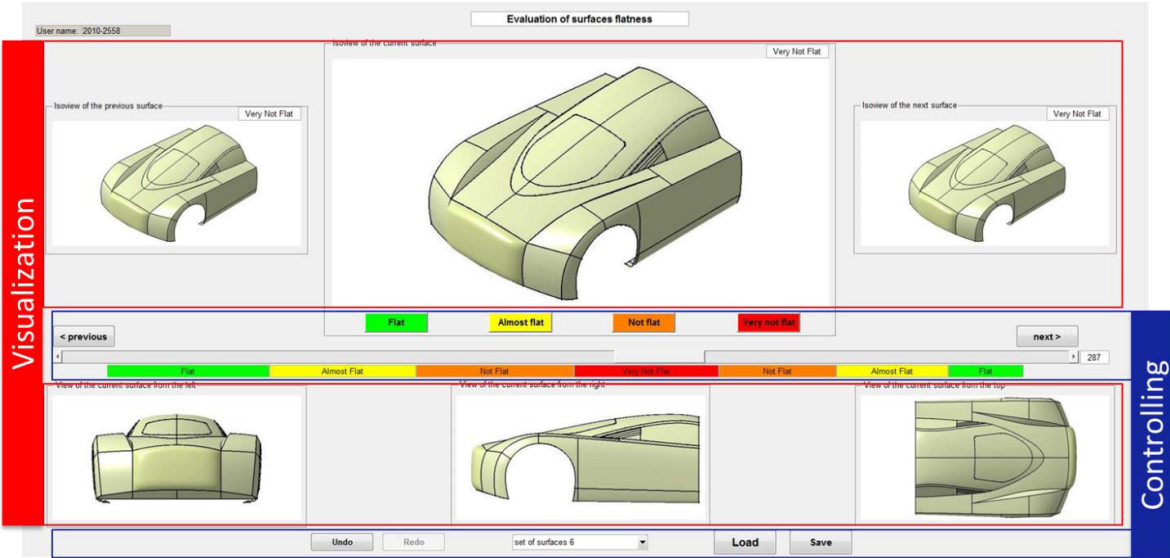
Figure 6.9: Workbench for the flatness classification of surface

The GUI is split into two parts: 1. Visualization part (red frames, Figure 6.9) and 2. Controlling part (blue frame, Figure 6.9). The visualization part contains a principal frame and a secondary frame. In the principal frame, the surface is displayed from the isoview. It contains three windows for displaying the surfaces. The middle one displays the current surface, which is the object of classification, while the other two display the previous (the left window) and the next surface (the right window). The display of the previous and next surfaces

is motivated by the possibility of having three neighbor surfaces in view, so participants are able to compare them and decide about the changes of the class. In the secondary frame, the surface is displayed from the other three standard points of view (view from the left, from the right, and from the top), helping the interviewees decide about the class of the current surface. The controlling part (bordered in blue) also contains the principal and secondary frames. The principal frame consists of a slider and four buttons. The four colored buttons correspond to one of the four proposed classification classes, so, for instance, by clicking on the green button, the surface is classified as flat and so on. The other two buttons allow moving to the previous and the next surface. In order to accelerate the classification process, the user does not have to classify every surface, but only those corresponding to a change of the actual class (last attribute selected). In other words, the interviewee classifies the first surface in one of the four classes, and then moves to the next surface. If he/she considers that the next surface also belongs to the same class as the previous one, then there is no need to click on the appropriate button; however, if the participant considers that the actual surface has changed the class, according to his/her perceptions, the class then he has to click on the corresponding button to record the change. In case the participant wants to revise or edit the classification, he/she can use the slider to move rapidly across the set of surfaces and change locally the class of few surfaces. The second frame of the controlling part serves for the manipulation of the classification process, in terms of selecting the set of surfaces (the 9 sets of surfaces, Figure 6.5). After the set of surface has been selected, by clicking on the Load button, they are loaded in the application to be classified by the participant. Next, after completing the classification process of the selected set of surfaces, the classification is saved in an appropriate text file by clicking on the button Save. In cases where the interviewee changes his/her mind about some classification and wants to repeat the classification of a given portion of surfaces, the classification can be repeated by clicking on the buttons Undo or Redo.

6.6   Experimentations

6.6.1   Organization of the Initial Data Set (IDS)

The classification has been conducted by interviewing participants from different countries (France, Italy, and Macedonia) and with different backgrounds (engineers, mathematicians, students, PhD candidates, researchers, and so on). The composition of the participants is given in Table 6.4.

Place	Number of participants	Status	Sex (M/F)	Age range	Common comments	Average duration
France	20	PhD student – 11 Masters – 4 Engineers – 5	15/5	23 – 33	Very well done Time-consuming	20 – 45 minutes
Italy	15	PhD degree – 7 Researcher – 7 Technical staff – 1	7/8	25 – 54	Good interface Too many views confuse the interviewees	35 – 60 minutes
Macedonia	30	PhD degree – 5 Engineers – 17 Students – 8	25/5	21 – 51	Interesting Time-consuming	30 – 50 minutes

After the interviews have been conducted and the classification data have been collected, the next step is to analyze and interpret the results. Before presenting the results, a repetition of the statement of the main hypothesis has to be made, accompanied by an expected outcome and followed by a foreseen activity to present the results. In terms of the interpretation of results, the entire dataset of instances (Figure 6.10) has been organized in a certain way, aiming at emphasizing the different aspects and influences to the perception of flatness.

The first and most important aspects of this work are to analyze the:

- Perception of flatness for every participant in the interviews

- Influence of the surrounding (context) on the perception of flatness

- Influence of different surroundings (objects) on the perception of flatness

- Perception of flatness of different groups of participants regarding their:

    - Background (Students, Engineers, and Researchers)

    - Provenance (France, Italy, and Macedonia)

    - Age and gender (up to 30 years old, and older than 30, male and female)
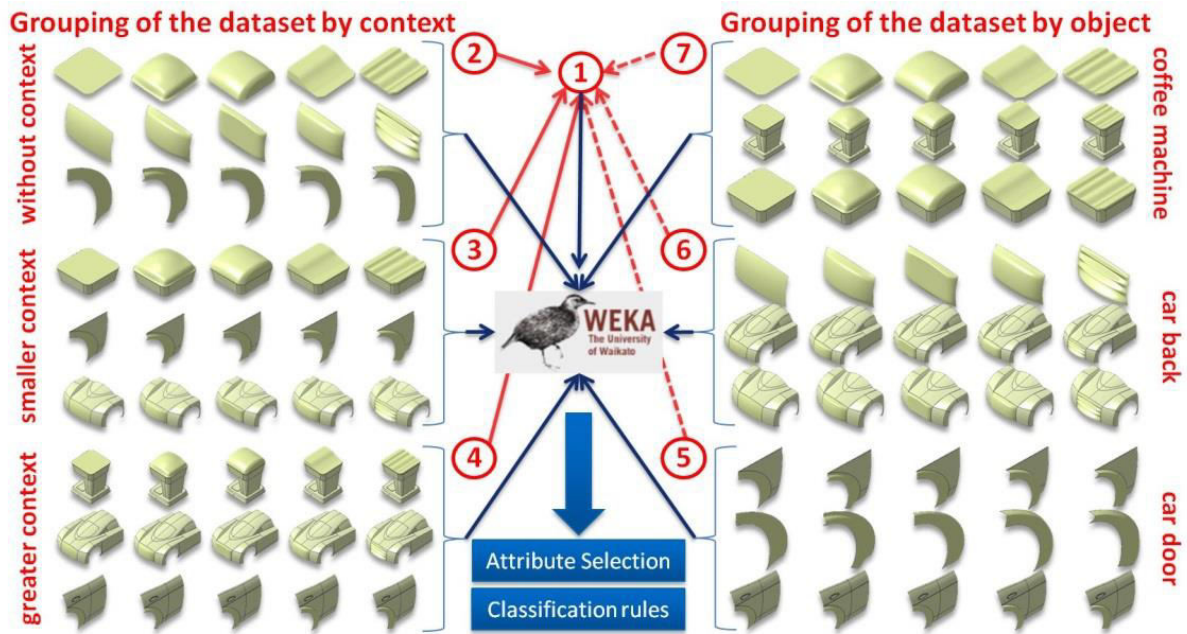
Figure 6.10: Organization of the initial dataset - IDS

The initial dataset IDS composed of 9 sets of surfaces (Figure 6.5) is organized in such a way that groups the sets of surface according to a certain property. To investigate the influence of the surrounding (context), the IDS is grouped in three groups, placing the sets of surfaces in a different group regarding whether the target shapes are given without context, with smaller context, or with greater context (Figure 6.10, left grouping). The motivation for this method of grouping is to examine what is the perception of flatness if the shape is given with or without context. In other words, the sets of surfaces 1, 3, and 8 (Figure 6.5) being merged in one group (Figure 6.10, group 2) would be perceived in one way, while the same shape placed in another context would be perceived differently. Our expectations are that if the shapes are given without any context, the participants will have a perception of flatness that differs much more (they will follow different rules of classification) from each other, resulting in lower classification accuracy. However, if the shapes are placed in an appropriate context (Figure 6.10, group 3 and 4), we expect that the participants will have an improved perception of flatness that will result in better classification consistency (they will follow similar classification rules) and higher classification accuracy.

Furthermore, in order to investigate the influence of the different surroundings, the IDS is also grouped in three different groups (Figure 6.10, right grouping): group 7 – coffee machine (sets of surfaces 1, 2 and 4); group 6 – car back (sets of surfaces 2, 6 and 7); and group 5 – car door (sets of surfaces 5, 8 and 9). The motivation behind this method of grouping is to examine what is the perception of flatness in different objects. Our expectation is that the perception of flatness has to be independent from the object where the shapes have been placed. This independence of the perception of flatness is followed by the fact

that the participant will follow the same classification rules regardless of whether the shape is placed in a coffee machine or in anything else. The comparison of the learning efficiency of all three groups (5, 6 and 7) will result in relatively constant classification accuracy.

Finally, the collection of all sets of surfaces in one single group (group 1, Figure 6.10) will represent how every single participant has perceived the flatness of the entire IDS. The application of the learning techniques will detect which classification rules have been followed by all participants. By doing so, we are able to detect if there is a common judgment of flatness.

6.6.2 Pre-processing of the acquired data from the classification

The fourth element of the framework is the beam of the temple, which represents the methods for pre-processing the acquired data and application of the adopted MLTs. The methods for pre-processing the data (for referential and mutual comparison) transform the multi-labeled classification problem into a single-label classification problem and verify the existence of a common judgment for flatness. The initial dataset IDS has been classified by all 65 participants, which means that all 8,550 instances in the dataset have been classified 65 times (multiple-label classification problems). In other words, 65 class labels have been assigned to each shape instance. Since the basic learning techniques can deal only with single-labeled classification problems, a method for transforming multiple-label classification problems into single-label classification problem has to be applied. The problem of dealing with multiple labeled dataset of instances can be solved in two ways: 1. replacing the multiple-labeled classifications with a single-labeled classification by using the majority principle and 2. application of Problem Transformation methods for solving multidimensional learning problems. In this thesis, the second approach for solving multiple-label classification problem has been used.

The objective of this work is to define whether there is a common judgment of flatness that includes, in the first place, an individual perception of flatness, and then to verify if all participants have followed more or less the same classification rules. If a general classification exists, it has to be extracted and evaluated. A separate comparison should be applied between the general classification and each classification of the participants.

Surface instances	classification by participants				classification classes				final class (>n/2)
	label 1	label 2	....	label 65	F	AF	NF	VNF	
surface 1	F	F	AF	NF	36	14	10	5	F
surface 2	F	F	F	F	30	25	6	4	
surface 3	F	F	AF	AF	15	38	8	4	AF
surface 4	AF	F	AF	AF	4	31	28	2	
surface 5	AF	NF	NF	AF	3	21	37	4	NF
surface 6	AF	NF	NF	NF	2	10	22	31	
....	...	...	...	...	...	...	...	...	
surface N	VNF	VNF	VNF	VNF	5	8	13	39	VNF

Figure 6.11: Assigning a final class to a surface using the majority principle

A general classification is a single (unique) classification model that replaces multiple individual classifications. In our case, the general classification represents the perception of flatness of the majority of participants. The creation of this model is made using voting principle so the overall single-label class that is assigned to a surface instance is same as the class assigned by the majority of participants. For instance, if one surface has been classified as flat by more than 33 out of 65 participants, then this class is delegated to be a final surface class (Figure 6.11). If there are no majority votes of same class to an instance, and assuming that the given surface instance will be classified same as one of the neighbor instances classified by the majority, then the comparison between the votes for these classes will decide on the final class (Figure 6.12). For instance, if we take surface 2 (Figure 6.11) as an example, it can be seen that this surface has been classified as flat by 30 participants, as an Almost Flat (AF) surface by 25 participants, as a Not Flat (NF) surface by 6 and Very Not Flat (VNF) surface by 4 participants. It is evident that this surface has not been classified in the same class by the majority of participants. The greatest number of classification in a same class is 30, with the class flat which is less than 33 (majority of 65), so to this surface no class can be assigned. One of the most important rules that have been followed during the generation of the sets of surfaces is the continuity in the modification of the shapes and creation of the intermediate surfaces. This means that if the surface k is classified as class1 and surface k+2 is classified as class2, then the surface in between (k+1) must belong either to class1 or class2. Therefore, if surface 2 cannot be delegated with a class by the majority of participants, then the surface should be classified same as some of the neighbor surfaces. Neighbor surfaces are considered the nearest (neighbor) surfaces that are classified with a final class by using the majority principle. In the example depicted in Figure 6.11, surface 1 is classified as F and surface 3 as AF, so surface 2 has to belong to either class A or AF. In order to determine which class (A or AF) has to be assigned, comparison between the votes associated to class A and AF is done, and the one with greater number of votes is chosen. In the given example, 30 participants have classified the surface 2 as flat and 25 as AF, so the final delegated class is F.

Surface instances	classification by participants				classification classes				final class (>n/2)
	label 1	label 2	....	label 65	F	AF	NF	VNF	
surface 1	F	F	AF	NF	36	14	10	5	F
surface 2	F	F	F	F	30 >	25	6	4	F
surface 3	F	F	AF	AF	15	38	8	4	AF
surface 4	AF	F	AF	AF	4	31 >	28	2	AF
surface 5	AF	NF	NF	AF	3	21	37	4	NF
surface 6	AF	NF	NF	NF	2	10	22 <	31	VNF
....	...	...	...	...	...	...	...	...	....
surface N	VNF	VNF	VNF	VNF	5	8	13	39	VNF

Figure 6.12: Assigning a final class to a surface regarding the neighbor surfaces

The goal of creating a general classification is to identify rules that reflect the perception of the majority of people. Technically speaking, this requires transformation of multiple-labeled instances into single-labeled surface instances. The only reasons why single-labeled surfaces are needed are due to the fact that the application of learning algorithms for creating classification models requires learning dataset of single-labeled instances.

The classification of the IDS by each of the participants is saved in a separate ARFF file obtaining 65 single-labeled datasets, making it suitable for applying of Attribute Selection techniques. Thus, the first method (CfsSubsetEval + BestFirst) has been applied to all the datasets extracting the subset of the relevant attributes for each participant. The attributes selected by the majority of the participants are considered as most relevant parameters of the surfaces, with respect to the flatness.

6.7   Results and discussion

6.7.1   Comparison of the learning capability of different learning algorithms

The evaluation of the capability of the learning algorithms to create an accurate classification model is done by using the 10-folds cross-validation method. From the interviews, we got 65 separate single-labeled datasets of surface instances. Each of these 65 single-labeled datasets has been used to create a classification model applying one of the five basic learners (previous chapter, Section 6.2) on which 10-folds cross-validations are applied to compute the accuracy of each classifier and the average learning accuracy. The same procedure is done to the other learning algorithms as well, and the results are given in Figure 6.13.

%	learners	C4.5	NaiveBayes	SMO	IBk	RIPPER
Single-labeled dataset	1. Participant classification	83,31	46,78	68,87	75,93	78,88
Grouping by context	2. Without context	81,62	49,08	69,57	74,81	78,68
	3. Smaller context	83,41	51,49	72,72	76,34	80,09
	4. Greater context	83,75	51,58	72,09	76,74	80,61
Grouping by object	5. Car door	76,99	57,48	72,63	66,30	74,94
	6. Car back	84,28	64,33	73,31	71,89	80,97
	7. Coffee machine	89,16	61,54	77,02	85,18	86,45

Figure 6.13: An average learning accuracy of different learning algorithms

From the obtained results, which are shown in Figure 6.13, it has been confirmed that the classification trees - C4.5 are very suitable in the type of applications where the attributes are numerical, which is the case in our application.

6.7.2 Perception of flatness of every participant in the interviews

The objective of this work is to investigate the perception of flatness by modeling this problem as Data Mining problem, so that we can apply MLTs to extract classification patterns and rules. As it was explained before, 65 persons have participated in filling out the questionnaire requesting them to classify a dataset of 8550 surfaces divided in 9 sets of surfaces. Furthermore, the investigation of existence of a common judgment for flatness can be carried out by:

1. Evaluation of the differences of the individual classifications with respect to the general classification model

This approach consists in first extracting a **general classification** model and then testing it with the classification of the interview participants to estimate how 'good' the general classification is. The dataset of surfaces classified by the general classification is used to train a classifier. Such trained classifier is tested by the same dataset but classified by the participants. The creation of the general classification is done using voting principle, so the overall single-label class that is assigned to a surfaces instance is the same as the class assigned by majority of participants (Figure 6.11 and Figure 6.12). The referent classification extracted by using the voting principle has been compared with the classification of all the participants in the interviews. The learning algorithm C4.5 (classification tree) has been used to train a classifier using the general classification (as a training set), and then the classifier is tested by the classification of all 65 participants (as a testing set). The comparison of the classification of all participants with respect to the general (referent) classification is given in Figure 6.14.

%	Participants	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Single label	Accuracy	67,7	45,0	43,6	53,6	40,5	35,9	48,4	68,4	49,6	58,1	57,3	57,0	45,8	59,1	50,1	47,3	51,2	63,1	39,0	50,1	51,2	71,7
	Ranking	3	51	54	29	58	64	45	2	43	21	22	26	48	18	42	47	36	10	59	41	37	1

%	Participants	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Single label	Accuracy	38,3	38,8	50,4	58,9	45,5	58,2	54,1	51,6	57,1	50,1	59,8	42,9	53,2	50,2	48,4	37,8	44,8	43,4	51,8	56,1	45,5	62,8
	Ranking	61	60	38	19	50	20	28	35	25	40	17	57	31	39	46	63	52	55	34	27	49	11

%	Participants	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	Av.
Single label	Accuracy	53,0	66,0	65,5	61,1	57,2	52,3	37,9	64,3	22,7	62,6	43,2	48,5	60,1	57,3	44,7	63,9	65,4	61,6	66,1	53,5	62,6	52,7
	Ranking	32	5	6	15	24	33	62	8	65	13	56	44	16	23	53	9	7	14	4	30	12	

Figure 6.14: Comparison between the classification of 65 interviewees and the one obtained using the general classification model

By analyzing the results given in Figure 6.11, the accuracy indicates the percentage of instances that have been classified in the same way by both the general classification model and the interviewee. Taking into account the accuracy of all participants, the average accuracy is 52.7%. If we rank the classification of all participants by accuracy and take the top-ten ranked classifications (red numbers), we can see that their average accuracy is 66.3%. Additionally, the highest accuracy (participant 22 – 71.7%, Figure 6.14) of the testing will point to the participant that shares more or less similar classification rules, with respect to the general classification.

2. Carrying out n*(n-1) mutual comparisons between the individual classifications

The idea is to find which the most shared classification for each surface is. What we want to examine is whether the classification rules that one person has followed are also recognized and shareable by the others. The most shared classification can be considered as relevant and final for the set of instances.

%	Training set						
	Participants	Participant 1	Participant 2	Participant 3	Participant 4	.....	Participant 65
Testing set	Participant 1	/	45,7	40,2	54,3	.....	59,8
	Participant 2	48,0	/	36,9	46,7	.....	44,5
	Participant 3	40,9	36,9	/	44,5	.....	44,8
	Participant 4	50,8	44,3	40,9	/	.....	57,6
	Participant 5	41,2	44,9	36,0	42,2	.....	40,9
	.....	.....	.....	.....	.....	.....	.....
	Participant 65	58,3	42,9	43,9	60,7	.....	/
Average accuracy		48,4	39,5	36,9	44,7	.....	47,3

Figure 6.15: Mutual comparison of individual classifications

Mutual comparison is carried out by using the classification of each participant to train a classifier (using C4.5). Then, this classifier is tested using the classifications of the other 64 participants, and the results of the testing are used for the comparison. This activity has been repeated for all the 64 participants, and the results are given in Figure 6.15. The average accuracy of the classification model for a participant can be considered as an overall and orientation measure for the "level of shareability" of the classification for this participant. The average accuracy is given in Figure 6.16. Taking into account the accuracy of all participants, the average accuracy is 42.1%. If we rank the classification of all participants ordered by the average accuracy and take the top-ten ranked classifications (red numbers), we can see that their average accuracy is 47.76%.

%	Participants	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Single label	Average accuracy	48,4	39,5	36,9	44,7	37,2	35,9	39,5	47,4	42,5	45,8	45,6	44,0	39,8	44,3	42,3	40,3	41,0	47,3	37,4	42,9	41,7	48,4
	Ranking	3	49	56	20	55	60	50	6	36	13	15	28	48	23	37	46	43	8	54	31	40	2

%	Participants	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Single label	Average accuracy	36,8	34,0	42,6	44,7	40,8	44,9	44,1	43,7	44,2	42,0	42,2	40,3	40,7	38,4	38,7	34,1	39,0	36,4	44,2	45,9	42,6	44,7
	Ranking	57	64	34	19	44	17	27	29	26	39	38	47	45	53	52	63	51	58	24	12	33	18

%	Participants	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	Av.
Single label	Average accuracy	41,1	46,3	46,3	47,6	44,2	43,5	34,6	49,2	26,4	45,4	34,2	41,4	42,6	44,5	36,2	47,4	45,7	44,6	47,9	42,8	47,3	42,1
	Ranking	42	10	11	5	25	30	61	1	65	16	62	41	35	22	59	7	14	21	4	32	9	

Figure 6.16: Mutual comparison classification model – the average accuracy of all participants

Having in mind the fact that mutual comparison indicates the classifications that are also recognized by the others, it can also help in defining which attributes are relevant in respect to the flatness.

An interesting conclusion can be drawn if we compare the top-ten ranked classifications of the referent and mutual comparisons (Figure 6.17). We can conclude that there is an 80% overlap in both comparison methods. Eight out of the first ten are shared between both methods (red frames, Figure 6.17). Since these eight participants have very high classification accuracy regarding the general classification (left array, Figure 6.17) and their classification rules are the most shared (right array, Figure 6.17) by the others, this leads to the conclusion that the general classification can be considered relevant.

referential comparisons

No	Participant	Accuracy	Rank
1	1	67,7	3
2	8	68,4	2
3	18	63,1	10
4	22	71,7	1
5	46	66	5
6	47	65,5	6
7	52	64,3	8
8	60	63,9	9
9	61	65,4	7
10	63	66,1	4

Average 66,3

mutual comparisons

No	Participant	Accuracy	Rank
1	1	48,4	3
2	8	47,4	6
3	18	47,3	8
4	22	48,4	2
5	46	46,3	10
6	48	47,6	5
7	52	49,2	1
8	60	47,4	7
9	63	47,9	4
10	65	47,3	9

Average 47,8

Figure 6.17: The ten highest accuracies result obtained with referential and mutual comparisons

6.7.3 Influence of the surrounding (context) to the perception of flatness

The main hypothesis is that if the surrounding does not affect the perception of flatness, the participants will follow the 'same' classification rules for the same surface regardless the surrounding. On the contrary, the participants will 'modify' their classification rules in case the surrounding has an influence on the perception of the surface shape.

%	Participants	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Single label	without context	64,2	44,9	30,1	49,4	37,7	38,6	46,9	67,6	48,3	52,6	59,8	63,4	38,9	54,6	49,4	37,6	46,2	59,8	37,2	50,8	51,7	70,0
	smaller context	70,2	42,9	38,0	56,6	43,9	33,1	44,2	65,6	48,5	69,7	57,5	59,9	44,8	60,8	48,4	52,3	50,2	58,3	34,0	52,8	49,9	72,1
	greater context	65,4	46,6	60,6	54,4	37,4	38,8	51,5	71,7	51,9	55,1	56,6	50,9	52,7	62,5	52,4	53,7	57,8	70,7	46,3	47,4	51,2	71,9

%	Participants	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Single label	without context	37,6	45,8	50,1	58,2	40,2	51,4	57,5	58,2	42,7	50,1	67,4	53,7	54,2	42,9	50,5	34,3	45,2	49,4	54,6	44,2	50,2	60,8
	smaller context	34,0	39,4	47,9	53,6	48,3	63,5	49,5	50,9	63,8	44,9	58,9	42,2	49,9	47,2	43,6	42,7	48,0	45,3	50,8	57,7	46,9	62,9
	greater context	40,6	33,0	52,8	65,5	45,7	59,8	55,4	48,1	59,5	56,4	54,1	35,7	55,3	57,6	48,4	36,5	40,8	35,2	54,2	66,4	41,3	64,3

%	Participants	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	Av.
Single label	without context	55,8	59,2	55,2	66,1	56,6	49,5	40,1	63,3	20,6	60,6	33,3	40,9	71,7	60,5	44,2	69,0	64,7	62,8	64,9	44,2	60,3	51,4
	smaller context	44,3	65,0	69,9	58,3	55,1	45,9	33,2	60,2	22,8	59,6	50,4	49,2	59,1	60,3	41,5	56,8	70,6	58,4	64,9	52,8	68,9	52,2
	greater context	60,7	74,5	71,5	60,0	60,8	59,1	39,1	70,1	23,5	66,7	44,6	52,0	50,1	51,7	49,1	68,5	62,1	63,1	69,0	63,1	58,9	54,3

Figure 6.18: Results of the comparison of the individual classification with the general classification model - influence of the surrounding

As mentioned before, to investigate if the surrounding influences the perception of flatness, the initial dataset IDS of surfaces is divided into three groups (groups 2, 3, and 4, Figure 6.10). Group 2 includes only the surfaces to be classified, while Groups 3 and 4 contain the same surfaces placed in an appropriate context (smaller and greater contexts, respectively, Figure 6.3). The general classification has been divided in the same way for the

three cases (without context, with smaller context, and with greater context). The investigation of the influence of the surroundings on the perception of flatness is done through the application of both the referent and mutual comparison methods. The referent comparison is carried out by testing the classification models created over all three groups of the general classification, with corresponding groups (without context, with smaller context, and with greater context) of each participant. Figure 6.18 shows the results for the accuracy obtained for the three sets. In the right red column (Figure 6.18), the average accuracy for each group of all participants is given. This result indicates that the average accuracy of the classification of the shapes without context is 51.4% and as the size of the context increases (smaller and greater context), the classification accuracy increases (52.2% and 54.3%, respectively). This order of accuracies confirms our hypothesis that the context influences perceptions of flatness in such a way that improves the perception of flatness and strengthens the classification consistency.

The same order of the classification accuracy is obtained applying the mutual comparison. The individual classifications of the three groups (2, 3 and 4) have been used to train the classifiers. Then, all three groups of instances have been used to test the three obtained classifiers. By repeating the same activity for all 65 participants, 65 matrices (3x3) have been obtained. The obtained results lead us to the conclusion that the accuracy of the classifier trained by group 4 is greater than the accuracy of the classifier trained and tested by group 3, which is in turn greater than the one trained by group 2. The average values are depicted in Figure 6.19.

Av	Training set		
Size of context	Without context	Smaller context	Greater context
Without context	86,51	45,91	47,47
Smaller context	58,65	88,26	55,16
Greater context	59,78	51,31	88,67
	43	41	52

Figure 6.19: mutual comparison classification model - influence of the surrounding

For instance, the accuracy of 86.51% (Figure 6.19) is an average value of the accuracy of testing the learned method from group 2 on the same group for all 65 participants. The results given in Figure 6.19 show that the average accuracy of the classifiers trained and tested by group 4 (88.67%) is greater than the average accuracy of the classifiers trained and tested by group 3 (88.26%). This order of the average accuracy corresponds to our statement that by increasing the context, the perception of flatness became more stable.

without context   smaller context   greater context

Figure 6.20: shape in different context

6.7.4 Influence of different surrounding (objects) to the perception of flatness

Since it has been confirmed that the context influences the perception of flatness, the next important step is to verify whether and how much the object type affects the perception of flatness. As was explained earlier, the research in this thesis has been conducted for three different objects: a coffee machine, a car, and a car door (Figure 6.1). It is true that the perception of flatness is influenced by the size of the surrounding, but it has to be independent from the shape of the surrounding. For instance, when someone is classifying a set of surfaces given in different size of surroundings of a same object (e.g. coffee machine), he/she will intuitively follow some classification rules. These classification rules do not differ greatly from the classification rules that he/she would follow when classifying other set of surfaces of other objects (car or car door). For this analysis, the initial dataset of 8,550 surfaces (Figure 6.5) is divided into three groups corresponding to different objects (groups 5, 6, and 7 in Figure 6.10), and the general classification obtained using the majority principle is also reorganized, corresponding to the three different objects.

%	Participants	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Single label	Car door	54,7	39,5	31,2	52,3	50,1	27,5	32,7	63,5	50,6	54,7	66,4	59,8	42,5	42,8	36,6	48,1	41,3	57,8	27,6	61,0	23,0	66,3
	Car back	73,1	45,8	39,4	49,9	28,9	32,5	53,4	80,4	39,4	60,3	43,9	49,0	61,6	73,2	44,2	39,1	56,1	58,7	32,0	42,3	60,3	82,3
	Coffee machine	73,3	49,0	57,8	57,4	39,6	50,9	56,8	61,1	60,0	64,4	64,2	65,3	32,5	61,8	69,1	56,4	56,3	73,2	58,1	48,5	68,7	66,3

%	Participants	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Single label	Car door	43,9	59,8	50,4	39,4	25,1	59,0	58,6	45,3	55,8	35,7	66,6	31,8	37,5	48,3	37,9	34,1	49,4	27,2	49,9	48,1	40,8	53,6
	Car back	25,6	16,8	51,0	73,4	49,6	55,0	54,4	47,1	46,4	49,1	75,7	33,6	53,8	60,1	59,1	45,4	51,1	45,5	46,8	54,0	40,9	80,3
	Coffee machine	44,0	41,4	49,6	66,0	61,1	62,4	50,8	64,9	64,6	65,6	38,9	66,0	69,1	38,8	45,2	34,7	34,2	56,6	63,4	66,6	56,1	57,1

%	Participants	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	Av.
Single label	Car door	54,6	49,1	66,7	62,2	46,4	37,0	25,8	59,2	15,2	51,9	43,4	30,9	62,9	43,8	72,5	61,5	65,3	55,6	71,5	50,4	54,6	47,8
	Car back	39,9	77,0	64,5	49,0	58,8	54,7	57,9	59,3	20,7	81,5	57,3	55,7	70,1	68,5	15,0	54,7	72,7	56,6	55,6	69,3	58,2	53,2
	Coffee machine	67,2	73,0	67,1	74,0	66,2	62,6	27,9	75,7	29,5	54,7	28,5	55,8	49,1	60,4	47,0	78,1	60,9	73,3	72,1	41,7	75,0	57,4

Figure 6.21: Results obtained with referent comparison - influence of different surrounding (objects)

Moreover, the investigation of the influence of the different object type on the perception of flatness is done using both the referent and mutual comparison methods. The

referent comparison is carried out by testing the classification models created, using the three groups of the general classification (extracted using the majority principle) with the corresponding groups (a coffee machine, a car, and a car door) of each participant. In other words, the general classification of the shapes of a specific group (e.g. group 7 coffee machine) has been used to train a classifier, and this classifier has later been tested on the classified shapes of the corresponding group (the coffee machine). Figure 6.21 shows the results obtained by applying this process on the three groups. For instance, the values 54.7%, 73.1%, and 73.3% correspond to the accuracies obtained by testing the three classification models created from the general classification on groups 5, 6, and 7, respectively, for the first participant. In (Figure 6.21), the rightest red column reports the average accuracy for each group of all participants. This result indicates that the average accuracy of the classification of the shapes for the car door is 47.8%, the average accuracy of the classification for the car back is 53.2%, and the accuracy of the classification for the coffee machine is 57.4%. The results given in Figure 6.21 show that there is a big difference between the average accuracies for the car door (47.8%) and the coffee machine (57.4%). The reason for obtaining these results lies in the fact that the enlargement of the surrounding affects the perception of flatness.

Av	Training set		
Objects	Car door	Car back	Coffee machine
Car door	81,79	29,86	28,37
Car back	36,00	89,53	30,03
Coffee machine	38,36	40,18	93,16
	50	43	48

Figure 6.22: mutual comparison - influence of different surrounding (objects)

Similar results have been obtained with the application of the second method – mutual comparison. In order to verify the consistency of each participant's own classification rules while classifying surfaces of different objects, all three groups of surfaces are used to train classifiers using the classification tree (J48) algorithm, and then the same groups are used as testing sets to evaluate their classification accuracy. In other words, to see whether a participant follows more or less the same classification rules while classifying the surfaces of a coffee machine and the surfaces of the car, a classification model for the group of coffee machine has to be created and then compared against all the surfaces of the other objects. Repeating the same procedure for the other two objects produces a matrix (3x3) of the classification accuracy of a participant. By repeating the same activity for all 65 participants, 65 matrices (3x3) will be obtained, and by computing the average values for each position of the matrices, the matrix given in Figure 6.22 is obtained. For instance, the accuracy value of 81.79% (Figure 6.22) is the average value of the accuracy of testing group 5 on itself for all 65

participants. The results given in Figure 6.22 show that the average accuracy of the classifiers trained and tested on group 7 (93.16%) is greater than the average accuracy of the classifiers trained and tested on group 6 (89.53%). The same conclusion can be drawn with the application of the first method – referent comparison. Finally, the results obtained by the application of both the referent and mutual comparison methods do not support our expectation, which means that the object does affect the perception of flatness (Figure 6.23).

Car door   Car back   Coffee machine

Figure 6.23: target shape in different objects

If we look at the results obtained through both referent and mutual comparison (Figure 6.21 and Figure 6.22) of the classification, we can see that the average accuracy of the classification for the car door is the lowest (47.8%), for the coffee machine is the highest (57.4%), and for the car is in between (53.2%). The same trend of the classification accuracy is also followed by the mutual comparison where the average accuracy of the car door is the lowest (81.79%), the average classification accuracy for the car is 89.53%, and the highest classification accuracy (93.16%) is for the coffee machine. If we analyze the results by comparing them with the corresponding object, we can conclude that the previously defined hypothesis has to be reformulated. Namely, we can argue that the position of the analyzed surface with respect to the surroundings affects the perception. In fact, the position of the surface regarding the surrounding affects the perception of flatness in terms of the shape transition towards the surrounding, and curvature and degree of continuity. For instance, the considered surface in the car door has an average normal vector (black arrow, Figure 6.24) that forms an angle β(smaller than 90°) when comparing the average normal vectors (blue arrows, Figure 6.24) of the surrounding surfaces. Having a surface positioned in surroundings that are more or less in the same plane (very small angle β) perturbs the perception of the shape from the participants. Furthermore, the surface in the car back is positioned in such way that its normal vector (black arrow) forms an angle β (smaller than 90°) with two near surfaces (blue arrows), and it also occupies an angle α(bigger than 90°) with two near surfaces (red arrows). Having surrounding surfaces positioned under a bigger angle improves the perception of the shape, which results in improving the classification accuracy (53.2%). Finally, the position of the target shape in the coffee machine (black arrow) occu-

pies an angle α (bigger than 90°), which helps the participant obtain a better view of the target shape, improving the perception of the shape and classification accuracy (57.4%).
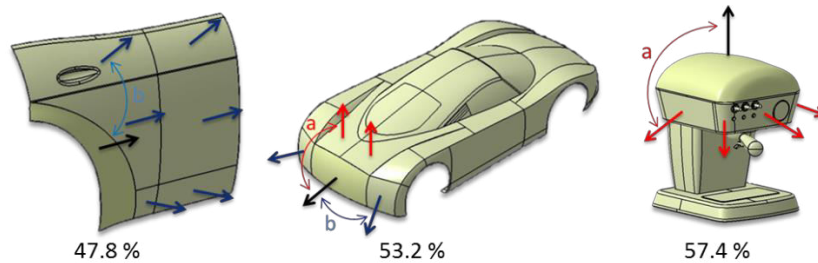

47.8 %   53.2 %   57.4 %

Figure 6.24: Positioning of the target shapes in the surrounding

6.7.5   Choosing the most relevant surface parameters

Since it has been confirmed that, during the interviews, all participants have followed, more or the same common rules, another objective of this work is to apply DM methodology to extract the most meaningful surface parameters for characterizing the flatness property. Finding out the most relevant parameters allows their use in determining an appropriate measure of a given aesthetic property, and it opens a door to its application in designing more user-friendly modeling tools. Such a modeling tool can allow the user to specify the changes on the flatness property providing the corresponding deformation of free-form surface. This capability spares the designers from manipulating surfaces with low-level geometric entities (points and curves) and provides shape-oriented deformation tools.

To solve the problem of identifying the important parameters for the flatness property, we adopted the Attribute Selection (AS), as a part of Data Mining methodology. AS allows identifying which attributes can be omitted without affecting the results of further analysis. As it was explained earlier in our work, the WEKA data mining workbench has been adopted. The WEKA provides two different methods for the AS regarding the attributes evaluation and their representation. The first method uses correlation based algorithm to evaluate a subset of attributes. Then, it applies appropriate search algorithms to rank and propose the best subset of attributes. In this case, the CfsSubsetEval – evaluation algorithm is used, and then BestFirst search algorithm is applied to propose a subset of attributes which is highly correlated with the classes, but the attributes in the subset are more independent from one another. The second method uses algorithms that provide independent evaluation of all attributes and then applies search algorithms to rank all attributes in a list. In this case, the InfoGainAttributeEval – evaluation algorithm is used to calculate the mutual information of the attributes and classes. Then, such calculated values are ranked in decreasing order by the Ranker algorithm.

The two methods for attribute selection have been tested so as to detect which one proposes a better solution or leads us to a unique set of relevant surface parameters. Having 65 separate single-labeled dataset of instance, requires utilization of the AS algorithms to each of the datasets. Each of the datasets represents the classification of a participant who has followed his own classification rules while classifying the surfaces. Therefore, the application of the first method of AS to a dataset will select those parameters that seem to be relevant for that participant. Repeating the same procedure for all 65 datasets, the AS algorithm selects 65 sets of parameters. We expect that, by counting the number of times that one parameter is selected from different participants, there will be a group of parameters that appear more often than others (Figure 6.25). As explained earlier, we defined 26 different geometric quantities characterizing both the analyzed surface (20) and the context (6) that were used for the specification of 36 size-independent (dimensionless) surface parameters. The red frames in Figure 6.25 indicate the surface parameters that have been selected by the algorithm for most of the datasets (more than 33 times). It is evident that only four parameters have been selected by the majority, but we cannot conclude that others cannot be considered relevant as well. This is justified by the fact that if we take into consideration the parameters ranked in the successive positions, we can conclude that parameters 17 (ranked as fifth) and 36 (ranked as sixth) were selected as relevant 31 times (which is only two less than the majority), while parameters 25 (ranked as seventh) and 27 (ranked as eighth) were selected as relevant 21 and 20 times, respectively. Therefore, we cannot simply use the majority because we risk omitting some possibly relevant surface parameters, meaning we need additional methods that will help us in the selection.

Single label	Parameters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	Selected	42	44	4	36	5	10	0	6	0	1	0	11	0	5	2	11	31	13
	Ranking	2	1	25	4	20	16	29	19	30	28	31	13	32	21	27	14	5	11

Single label	Parameters	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
	Selected	5	8	0	0	0	0	21	4	20	41	17	5	16	5	7	12	11	31
	Ranking	22	17	33	34	35	36	7	26	8	3	9	23	10	24	18	12	15	6

Figure 6.25: Relevant attributes of all participants

Another method that can help us to identify the most relevant surface parameters is general classification. As explained before, a general classification has been extracted from the multiple-label classification obtained from the classification of 65 participants, transforming (by voting majority) the multi-label classification into a single-label classification. Since the general classification represents (in some way) the classification of all participants, a set of attributes has been selected (Figure 6.26) by applying the same relevant attribute-selection algorithms (CfsSubsetEval + BestFirst). By analyzing the results given in Figure 6.26, it can be concluded that not only the parameters (1, 2, 4, and 28, Figure 6.25) were selected

by the voting majority principle, but also parameters 8, 17 and 36 (blue frames, Figure 6.25), were selected too.

index	Surface parameters
1	Ratio between (As/Ap)
2	Ratio between (Vs/V)
4	Ratio between (E2/D)
8	Ratio between E3/E1 (E3<E1)
17	Mean curvature (Mc*Vs/Ap)
28	Average normal |Na|
36	Distribution of the normal $\frac{1}{k}\sum_1^k \frac{dot(N_a,N_{aoi})}{\|N_a\|\,\|N_{aoi}\|}$

Figure 6.26: Relevant attributes of the general classification model

Both the referent and mutual comparison methods have an 80% overlap of the top-ten best-ranked classifications, which means that eight out of ten are the same (Figure 6.17). Since the 12 best ranked participants (8 common + 2 remaining from the first method + 2 remaining from the second method) have very high classification accuracies regarding both the general classification (52.3%, Figure 6.27) and the mutual comparison accuracy (58.8%, Figure 6.27), this leads to the conclusion that their classification rules are shared by the majority. Therefore, by applying the same relevant attribute selection algorithms (CfsSubsetEval + BestFirst) to all 12 classifications, 12 sets of attributes have been selected. By counting the number of times that one parameter is selected from the 12 sets of attributes, a list of parameters selected by the majority (more than 6) is created. This list contains 13 parameters, and it means that the set of relevant parameters will be identified among these parameters.

Number of parameters	Number of participants	methods	train/test validation (%)
36	65	mutual	42,11
		referential	52,67
	12	mutual	58,80
		referential	52,25

Figure 6.27: Classification accuracy of the 12 best ranked classifications

The identification of such set of parameters can be considered as a potential candidate for relevant surface parameters. To be able to state that this group of surface parameters is sufficient, after the less relevant parameters are omitted, the accuracy of the classification model of the datasets has to increase (or at least remain the same). That is to say, the sufficient surface parameters are those for creating the classification model, so the classification will remain unchanged after removing the less relevant parameters. The results in Figure

6.28 state exactly that. Namely, after removing the less relevant surface parameters, the classification accuracy using both the referential and mutual comparison methods has been computed over the classification of all 65 participants and the 12 best ranked.

Number of parameters	Number of participants	methods	train/test validation (%)
13	65	mutual	42,14
		referential	52,70
	12	mutual	56,47
		referential	52,29

Figure 6.28: Classification accuracy after omitting less relevant parameters

It is very interesting to compare the classification accuracy of the application of both the referential and mutual comparison methods for the 65 participants in cases where the number of parameters is 36 (Figure 6.27) and 13 (Figure 6.28). Their accuracies remain almost the same (or they differ by 0.03%), which confirms the expectation that removing the less relevant parameters would not significantly modify the accuracy. Having almost the same classification accuracy indicates that the less relevant or irrelevant parameters have been identified, but it does not point at the most relevant parameters among them. The best 10 ranked parameters from the Figure 6.25 and the parameters given in Figure 6.26 do not provide a unique list of parameters, and there are some parameters in the first figure that do not appear in the second figure and vice versa. Therefore, an additional and more exhaustive investigation is required, taking into consideration the classification rules followed by the 12 best-ranked participants. Since we decided that the 13 parameters are relevant, now we have to identify the most relevant parameter among them. In order to do that, the less relevant parameters (23) are removed from all 65 datasets of instances (filtering the parameters), and the same procedure for relevant attribute selection is repeated. Repeating the same procedure for all 65 datasets, the AS algorithm will select 65 sets of parameters for each dataset, respectively. As before, by counting the number of times that one parameter is selected from different participants, there will be group parameters that will appear more often than others (Figure 6.29).

Rank	Surface Parameters	Selected	index
1	Ratio between (Vs/V)	45	2
2	Ratio between (As/Ap)	42	1
3	Ratio between (E_2/D)	42	4
4	Average normal |Na|	39	28
5	Distribution of the normal	34	36
6	Mean curvature (Mc*Vs/Ap)	32	17
7	Positive curvature (Np)	26	25
8	Ratio between D/ (Do+D)	24	35
9	Zero curvature (Nz)	19	27
10	Ratio between (Vs/Vo+Vs)	18	31

Figure 6.29: Ranking of the relevant parameters

Furthermore, after counting the number of times that one parameter is selected from different participants, Figure 6.29 demonstrates the surface parameters selected by the majority (more than 32 times). Analyzing the parameters (Figure 6.29) selected by the majority, it can be concluded that they are exactly the same parameters selected by all 65 participants among 36 parameters (the top-six ranked parameters, Figure 6.25) and parameters selected by the general classification (excluding parameter 8, Figure 6.26). Finally, using the application of our methodology, six surface parameters have been selected (parameter 1, 2, 4, 17, 28, and 36, Figure 6.30).

index	Selected parameters
1	Ratio between (As/Ap)
2	Ratio between (Vs/V)
4	Ratio between (E_2/D)
17	Mean curvature (Mc*Vs/Ap)
28	Average normal |Na|
36	Distribution of the normal

Figure 6.30: Selection the most relevant parameters

6.8 Conclusion

This chapter describes the application of the framework to the detection of classification patterns of 3D free-form surfaces, with respect to the aesthetic property – Flatness. The objective of this chapter is to investigate whether there is a common judgment for the flatness perceived by non-professional designers, people that can be considered as potential customers. It is evident that the perception of flatness (or any other aesthetic property) is not affected only by the classified surface shape itself, but also by the surrounding (i.e. the size of the surrounding) and the shape transition towards the surrounding. Unlike the straightness of curves, the application of the framework for surface requires facing much more challenges and answering many questions. The first and most crucial part of the framework is the generation of the Initial Dataset of instances. We decided to apply morphing mechanism to obtain a wide set of surfaces, thus the key elements of the IDS are the definition of the target shapes and their ordering in a Deformation Path. The selection of the target surfaces has to be done carefully because their geometric characteristics (e.g. area, curvatures, minimal bounding box and so on) will directly affect the perception of flatness. Therefore, the selected target shapes have to contain certain geometric characteristics aiming at investigating their influence to the perception of flatness. In order to investigate the influence of the surrounding and the transition of the shape towards the surrounding to the perception of flatness, the surfaces created by morphing the Deformation Path are placed in appropriate objects.

The second element of the framework refers to the definition of geometric quantities related to the surface and the surrounding. Since the flatness of surface in 3D space is an extension of the curve straightness in 2D, it is reasonable to use the same geometric quantities for straightness, but extended in 3D, as a base for defining the geometric quantities for surfaces – Attributes. The construction of the surface parameters has to satisfy two requirements: 1. to be as much as possible relevant in respect to the flatness, and 2. to be dimensionless. The third element of the framework is also a very challenging task. Namely, this task consists of classifying 8550 surface instances, which is very tedious activity if the classification is carried out by classifying these instances printed on a piece of paper. Therefore, a GUI in Matlab has been created to facilitate the classification process and to reduce the classification time up to a few tenth of minutes. The participants were requested to classify the IDS in four classes (Flat, Almost Flat, Not Flat, and Very Not Flat) only by moving a slider and clicking on four buttons. Furthermore, having on the one hand the surface parameters (36), and on the other hand the classification carried out by a group of non-professionals, the IDS dataset is ready to be used for the experimentations. In order to investigate the different aspects of the perception of flatness, the IDS dataset has been organized in two ways: 1. the IDS is grouped in three groups according to the size of the surrounding (without, smaller, and greater context), and 2. the IDS is grouped in three groups according to the objects which the surfaces are placed in. The adopted learning algorithms are used for solving sin-

gle-labeled classification problems, while the classification of surfaces by interviewing people creates multi-labeled classification problems. Therefore, by using the majority principle, the acquired classifications have been processed transforming them into single-labeled classification (i.e. general classification). To investigate the existence of a common judgment for the flatness, two methods have been applied: 1. by evaluating the general classification model while using the individual classification of the participants – referential comparison, and 2. by carrying out n (n-1) mutual comparison between the individual classification – mutual comparison. The results from the first method show that the average accuracy of the general classification, tested by all participants, is 52.7%, whereas the highest accuracy of a participant is 71.7%, i.e. 66.3%, considering the top-ten ranked classification. In the mutual comparison, the average classification accuracy considering all participants is 42.1%, whereas the highest accuracy is 49.2%, i.e. 47.76%, considering the top-ten ranked classification. Furthermore, the investigation of the influence of the surrounding shows that the greater the surrounding, the better the perception of flatness, and this strengthens the classification consistency. The results from the first method (referential comparison) demonstrate that the average classification accuracy of the shapes without context is 51.4%. By increasing the size of the context (i.e. smaller or greater), the classification accuracy increases too (52.2% or 54.3%, respectively). The same order of the accuracies is obtained when applying the second method – mutual comparison: 86.51% - without context, 88.26% - smaller context, and 88.67% - greater context. By considering both referential and mutual comparison, it can be confirmed that the increase of the context size does improve the perception of flatness and strengthens the classification consistency. The next important aspect is to verify whether and how much the object type affects the perception of flatness. The results obtained by applying both referential and mutual comparison, demonstrate that the average classification accuracies for the car door is the lowest (47.8% - referential comparison and 81.79% mutual comparison), for the coffee machine is the highest (57.4% - referential comparison and 93.16% mutual comparison), and for the car is in between (53.2% - referential comparison and 89.53% mutual comparison). These results confirm that the shape transition towards the surrounding affects the perception in such way that if there is an abrupt change in the shape surround, the perception of flatness is improved. Finally, the application of Attribute Selection algorithms (evaluation and search algorithms) makes us succeed in the identification of the most relevant surface parameter. Using the application of our methodology, six surface parameters have been selected (parameter 1, 2, 4, 17, 28, and 36Figure 6.30). The first in the list of selected relevant parameters is the parameter 1, i.e. ratio between A_s and its projection A_p onto biggest Minimal Bounding Box (MBB) plane. The second parameter is the ratio between surface volume V_s and the volume V of MBB. The third parameter is the ratio between second longest edge E_2 and the diagonal D of the MBB. The fourth parameter refers to the Mean curvature, multiplied by the ratio between the surface volume V_s and the surface projection area A_p. The fifth parameter takes into account the distribution of the normal (N_a) at the surface discretization points, and the sixth parameter considers its distribution (orientation) to the surrounding normal (N_{aoi}).

Conclusion and Perspectives

7 Conclusion and Perspectives

The long-term objective of this work is to define a methodology that will efficiently integrate concepts from the Affective Engineering (AE) into the Product Development Process. AE aims at providing an efficient platform to directly map surface shapes with aesthetics and emotions, thus allowing the incorporation of emotional features into appealing products. Designing appealing objects plays a key role in the commercial success of a product. Therefore, it would be very interesting to be able to design shapes by manipulating their aesthetic properties while controlling the evoked positive emotions. Actually, such new capabilities represent the first steps for defining a new geometric modeling approach that could be put in the category of Declarative Modeling (DM) approaches. The main advantage of DM is the ability to create objects by providing a simple description (e.g. set of abstract words) that can be widely known among non-professional designers.

Contributions of the thesis

As a first step towards the integration of AE in the Product Development Process, the objective of this work is to discover interesting relationships and rules between aesthetic properties of shapes and their geometric representations.

In the first part of this document, the concepts used as a basis of the proposed approach are presented together with the limitations of the existing approaches that have motivated this work. Therefore, the first set of concepts and models presented in part A refers to the basic geometric models. Then, methods addressing the mapping of aesthetic properties with shapes are introduced and discussed. Since the proposed approach is using Machine Learning Techniques to find such a mapping, those techniques are also presented and discussed.

In the second part, the contributions are presented and detailed. First, the proposed generic framework for mapping aesthetic properties to free-form shapes is introduced. It uses Machine Learning Techniques to extract aesthetic classification rules linking aesthetic

shape properties with geometric parameters. The proposed framework is tested on 2D curves with a focus on the straightness, which is an important property in the automotive industry. The proposed framework is then extended to free-form surfaces to discover relationships and rules linking the flatness of a surface to its geometric quantities. Once those rules are obtained, it is possible to make use of them to classify new unknown shapes.

A generic framework

Bridging the gap between the geometric quantities characterizing a shape and its aesthetic properties is carried while using Machine Learning Techniques (MLTs) incorporated in a generic framework. Our generic framework consists of 5 elements forming a temple, where the basis is represented by the initial dataset of instances (IDS). The use of MLTs is represented by the temple beam which is supported by two pillars. The first pillar gathers together the geometric quantities associated to each geometric entity (either a 2D curves or a free-form surface in our context) of the entire IDS. The second pillar gathers together the classification associated to each element of the entire IDS, with respect to aesthetic properties (either the straightness or the flatness in our context). The final (fifth) element of the temple is the roof that represents the results given in the form of classification rules, gathering together the list of meaningful attributes involved in the rules, the type of adopted classifier as well as its parameters.

Validation on 2D free-form curves

This framework has been validated first on curves and its aesthetic property, namely the straightness. At this stage, the validation process consists of testing the capability of MLTs for discovering structural classification patterns and identifying the most relevant geometric quantities with respect to the straightness. The results of the validation process show that MLTs can correctly classify 99.78 % of all curve instances, while the relevant attribute selection has chosen exactly the same curve parameters that were used in the computation of the measure of straightness. Therefore, the work presented in Chapter 2 is considered to be a first step towards the characterization and classification of free-form surfaces with respect to their aesthetic properties.

Extension to free-form surfaces

Unlike the curves, the surfaces are much more complex geometric entities and the framework application requires facing several challenges that have been presented in Chapter 2. Despite applying the framework for mapping the surface flatness to the free-form surface shapes, this thesis also investigates two other aspects involved in the perception of

flatness. First, how does the surrounding (context) can influence the perception of flatness? Second, how do the different surroundings (objects) influence the perception of flatness?

The results from the investigation, i.e. whether there exists a common judgment for the flatness perceived from non-professionals designers, show that the average classification accuracy of the general classification is 66.3%. The mutual comparison between classification of all participants in the interviews shows that the highest classification accuracy is 48.4%. Regarding the additional aspects of the investigation of the perception of flatness, it is indicated that the greater the surrounding, the better the perception of flatness, and this improves the classification consistency. Furthermore, the shape transition towards the surrounding, in terms of position and curvature continuity, also affects the perception of flatness.

Considering the second task of the framework application, i.e. the identification of relevant surface geometric quantities involved in the flatness characterization, we succeeded in identifying the most relevant surface parameters.

Here, it is very important to underline that this thesis does not tend to propose exact classification rules with respect to the flatness, but to demonstrate that the proposed framework can drive the investigation to satisfactory results. It is evident that the extracted general classification is highly correlated and very dependent on the selected objects, target shapes, and defined deformation path, as suggested in this work. Other relevant factors that also affect the final results of the framework are the profile and the background of the participants involved in the interview. The framework gives an opportunity for modifying or upgrading any of its elements and running the entire procedure again for getting new results, i.e. new classifiers together with their control parameters and new meaningful attributes. For instance, the initial dataset can be enriched by adding another set of surface instances, classified according to our classification classes. After running the entire framework one more time, the general classification will take into account the newly added surfaces and will suggest improved classification rules. This is of a great interest when considering knowledge capitalization. Another improvement of the classification rules can be obtained if, for instance, we provide classification of the same IDS by carrying out interviews on an additional group of participants. Such obtained classification is integrated in the third element (the pillar 2) and the framework is run again.

The perception of flatness has been also investigated from the aspects of influence of the surroundings and the shape transition towards the surroundings. Based on the results, it can be concluded that the perception of flatness cannot be considered as an intrinsic property of the given shapes, but it is highly affected by the surrounding, its position (i.e. the transition towards) in the surrounding, as well as the application domain. Therefore, when integrating the concept of aesthetic property, it is mandatory that the surrounding of the shape is taken into consideration.

The application of the framework also gives results by selecting a set of surface parameters that can be considered relevant. We succeeded in identifying six surface parameters and these surface parameters, among the set of 36 parameters, are the most relevant with respect to the flatness. Again, we cannot state that the selected surface parameters are generally the most relevant. They are highly dependable on the dataset of instances, on the target shapes that have been used, on the type of classifier and so on. Our framework provides opportunity for someone else to compute or present different types of surfaces parameters. Such presented surface parameters are integrated in the second element (the pillar 1) and the framework is running again to identify, among the presented parameters, which are the most relevant.

Perspectives

Discovering rules associated to other aesthetic properties

The aim of the framework is to establish a structure for mapping aesthetic properties to free-form shapes and by modifying any of its constitutive elements (i.e. base, beam, pillars and roof) to change the final results (i.e. the configured classifier). The same framework can also be applied for mapping other aesthetic property than flatness, e.g. the tension, the convexity of a surface. In general, the proposed framework can be used as a guided path for identifying a mapping between different semantics and free-form shapes.

Moving from the aesthetic to the emotional level

Thus, in the future, the same framework should be also applied for direct mapping between emotions and free-form shapes. As mentioned earlier, the application of this framework is for solving supervised learning problems. So, instead of using supervised learning, the learning algorithm in the fourth element (the beam) can be replaced with an unsupervised learning algorithm, such as clustering. This alternative use of the framework can help in clustering the IDS and the attributes in order to find the same internal correlation between instances. Thus, categories of emotional impacts could be discovered.

Linking identified rules with geometric deformation operators

In the future, we also want to make use of this work and the associated rules, to implement new geometric deformation operators that could deform free-form curves and surfaces through an appropriate vocabulary. This has already been tested for 2D curves. In this case, the rules are transformed in a set of geometric constraints and minimization functions to be solved. Thus, an extension of such operators to free-form surfaces can be thought.

Finally, the framework is very generic and can be applied not only for finding mappings between semantics and free-form shapes, but it can also be applied in other research fields besides Affective Engineering and industrial design.

8 References

11Ants Analytics. (2014, April 14). (Research & Engineering Centre) Consulté le March 20, 2015, sur http://www.11antsanalytics.com/

Abreu, M., & Fairhurst, M. (2008). *An Empirical Comparison of Individual Machine Learning Techniques in Signature and Fingerprint Classification.* Springer-Verlag Berlin Heidelberg.

Aher, B. S., & Lobo, L. M. (2011). Data Mining in Educational System using WEKA. *International Conference on Emerging Technology Trends (ICETT).*

Akadi, E. A., Ouardighi, E. A., & Aboutajdine, D. (2008). A Powerful Feature Selection approach based on Mutual Information. *International Journal of Computer Science and Network Security, 8*(4), 116-121.

Almuallim, H., & Dieggerich, T. (1994). Learning Boolean Concepts in the Presence of Many Irrelevant Features. *Artificial Intelligence , 69*, 279-306.

Almuallim, H., & Dietterich, G. T. (1992). Efficient Algorithms for Identifying Relevant Features. *Ninth Canadian Conference on Artificial Intelligence.*

Ankerst, M., Kastenmoller, G., Kriegel, H.-P., & Seidl, T. (1999). Nearest Neighbor Classification in 3D Protein Databases. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology.*

Atmosukarto, I. (2010). *3D Shape Analysis for Quantification, Classification, and Retrieval.* PhD Thesis: University of Washington.

Ault, K. H. (1999). 3D Gemoetric Modeiling for the 21st Centry. *Engineering Design Graphics Journal, 63*(2), 33-42.

Bajaj, L. C., Chen, J., & Xu, G. (1994). Free Form Surface Design with A-Patches. *Graphics Interface*, (pp. 174-181).

Barbier, A., Galin, E., & Akkouche, S. (2004). Complex Skeletal Implicit Surfaces with Levels of Detail. *Journal of WSCG'2004, 12*(1-3).

Bardis, G., Makris, D., Golfinopoulos, V., Miaoulis, G., & Plemenos, D. (2011). A Formal Framework for Declarative Scene Description Transformation into Geometric Contraints. Dans A. König, A. Dengel, K. Hinkelmann, K. Kise, J. R. Howlett, & C. L. Jain, *Knowledge-Based and Intelligent Information and Engineering Systems, Part I* (pp. 347-356). Kaiserslautern, Germany: Springer-Verlag Berlin Heidelberg.

Barutcuoglu, Z., Schapire, E. R., & Olga, T. G. (2006). Hierarchical multi-label prediction of gene fun. *Bioinformatics , 22*(7), 830-836.

Bielza, C., Li, G., & Larranaga, P. (2011). Multi-Dimensional Classification with Bayesian Networks. *International Journal of Approximate Reasoning, 52*, 705-727.

Blockeel, H., Raedt, D. L., & Ramon, J. (1998). Top-down induction of clustering trees. *Proccedings of the 15th International Conference on Machine Learning.*

Bloomenthal, J. (1999). *Implicit Surfaces.* Seattle: Marcel Dekker.

Boutell, R. M., Luo, J., Shen, X., & Brown, M. C. (2007). Learning multi-label scene classification. *Pattern Recognition, 37*(9), 1757-1771.

Bramer, M. (1999). *Knowledge Discovery and Data Mining.* London: Institution of Electrical Engineers.

Caruana, R., & Freitag, D. (1994). Greedy Attribute Selection. *Machine Learning: Proceedings of the Eleventh International Conference.*

Caruana, R., & Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. *International Conference on Machine Learning.* Pittsburgh.

Catalano, E. C., Ivrissimthis, I., & Nasri, A. (2007). Subdivision surfaces and applications. Dans M. Spagnuolo, & L. De Floriani, *Shape Analysis and Structuring* (pp. 115-143). Berlin: Springer.

Chasen, S. (1996). *A little history of C4 - the CAD/CAM handbook.* New York: McGraw-Hill.

Chauvat, D. (1994). *Le projet VoluFormes : un exemple de modélisation déclarative avec controle spatial, PhD thesis.* Nantes: University of Nantes.

Clare, A., & Ross, K. D. (2001). Knowledge discovery in multi-label phenotype data. *Proceedings of European Conference on PKDD.*

Cohen, W. W. (1993). Efficient pruning methods for separate-and-conquer rule learning systems. *Proceedings of the 13th International Joint Conference of Artificial Intelligence.* Chambery, France.

Cohen, W. W. (1995). Fast Effective Rule Induction. *Machine Learning: Proceeding of the Twelfth International Conference .*

Cohen, W. W., & Singer, Y. (1999). A simple, fast, and effective rule learner. *Proceedings of Annual Conference of American Association for Artificial Intelligence.*

Comité, D. F., Gilleron, R., & Tommasi, M. (2003). Learning multi-label alternating decision trees from texts and data. *The 3rd International Conference on Machine Learning and Data Mining Pattern Recognition.*

Conilione, C. P., & Wang, D. (2011). Automatic localization and annotation of facial features using machine learning techniques. *Soft Comput, 15*(6), 1231-1245.

Crammer, K., & Singer, Y. (2003). A family of additive online algorithms for category ranking. *Journal of Machine Learning Research, 3*, 1025-1058.

Danglade, F. P.-P., & Veron, P. (2014). On the use of machine learning to defeature CAD models for simulation. *Computer-Aided Design and Applications, 11*(3), 358-368.

Daniel, M., & Lucas, M. (1997). Towards Declarative Geometric Modelling in Mechanics. Dans *Integrated Design and Manufacturing in Mechanical Engeneering* (pp. 427-436). Kluwer Academic Publishers.

Dankwort, W., & Podehl, G. (2000). A New Aesthetic Design Workflow - Results from the European Project Fiores. Dans *CAD Tools and Algorithms for Product Design* (pp. 16-30). Springer.

Dash, M., & Liu, H. (1997). Feature Selection for Classification. *Intelligent Data Analysis 1*, 131–156.

Dembczynski, K. C., & Hullermeier, E. (s.d.). Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. *27th International Conference on Machine Learning.* Haifa.

DeRose, T., Kass, M., & Truong, T. (1998). Subdivision Surfaces in Character Animation. *Proceedings of SIGGRAPH 1998.*

desJardins, M., Eaton, E., & Wagstaff, L. K. (2006). Learning User Preferences for Sets of Objects. *Proceedings of the 23rd International Conference on Machine Learning.* Pittsburgh.

Desmontils, E. (1995). *Les modeleurs déclaratifs.* Nantes: RR-IRIN-95.

Dunham H., M. (2003). *Data Mining: Introductory and Advanced Topic.* Prentice Hall/Pearson Education.

Ebert, S. D. (1997). *Advanced Geometric Modelling.* CRC Press.

Ebert, S. D., Musgrave, F. K., Darwyn, P., Perlin, K., & Worley, S. (2003). *Texturing and Modeling: A Procedural Approach.* Morgan Kaufmann.

Ebert, S. D., Rohrer, R., Shaw, D. C., Panda, P., Kukla, M. J., & Roberts, D. A. (2000). Procedural shape generation for multi-dimensional data visualization. *Computers & Graphics, 24*, 375-384.

Eckel, G., & Jones, K. (2004, 12 07). *techpubs library*. (Silicon Graphics) Consulté le 04 18, 2015, sur http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=0650&db=bks&srch=&fname=/SGI_Developer/Perf_PG/sgi_html/ch09.html

Edelstein, A. H. (1999). *Introduction to Data Mining and Knowledge Discovery.* Potomac, USA: Two Crows.

Farin, G. (1996). *Curves and Surfaces for Computer Aided Geomtric Design: A Practical Guid.* Academic Press.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. *American Association for Arificial Intellegence*, 1-34.

Ferreira, J. A., & Figueiredo, A. T. (2012). Efficient feature selection filters for high-dimensional data. *Pattern Recognition Letters, 33*, 1794-1804.

FIORES. (1997). *Formalization and Integration of an Optimized Reverse Engineering Styling Workflow.* www.fiores.com: Project N. BE96-3579.

FIORES-II. (2000). *Character Preservation and Modeling in Aesthetic and Engineering Design.* http://www.fiores.com/: Project N: GRD1-1999-10385.

Flores, J., Gamez, J., & Martinez, A. (2014). Domains of competence of the semi-naive Bayesian network classifiers. *Information Sciences, 260*, 120-148.

Foley, D. J., van Dam, A., Feiner, K. S., & Hughes, F. J. (1992). *Computer Graphics: Principles and Practice.* Addison-Wesley.

Fontana, M., Giannini, F., & Meirana, M. (1999). A Free Form Feature Taxonomy . *Eurographics'99, 18*(3), 107-118.

Fontana, M., Giannini, F., & Meirana, M. (2000). Free From Feature for Aesthetic Design. *International Journal of Shape Modeling, 6*(2), 273-302.

Furnkranz, J., Hullermeier, E., Mencia Loza, E., & Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning, 73*(2), 133-153.

Ganster, B., & Klein, R. (2007). An Integrated Framework for Procedural Modeling. *Spring Conference on Computer Graphics*, (pp. 150-157). Bratislava.

Garner, R. S. (1999). WEKA: The Waikato Environment for Knowledge Analysis. Waikato, Hamilton.

George, J. H., & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. *11th Conference on Uncertainty in Artificial Intelligence.* San Mateo.

Geuzuaine, C., Marchandise, E., & Remacle, J.-F. (1999). *An introduction to Geometrical Modelling and Mesh.* The Gmsh Companion.

Ghiselli, E. E. (1964). *Theory of Psychological Measurement .* New York: McGraw-Hill.

Giannini, F., & Monti, M. (2002). An innovative approach to the aesthetic design. *Common Ground , Design Research Society, International Conference.* London.

Giannini, F., & Monti, M. (2002). CAD Tools Based on Aesthetic Properties. Dans *Eurographics Italian Chapter.* Milano: Facolta' del Design - Politechico di Milano.

Giannini, F., & Monti, M. (2003). Design intent-oriented modelling tools for aesthetic design. *Journal of WSCG, 11*(1).

Giannini, F., & Monti, M. (2010). A survey of tools for understanding and exploiting the link between shape and emotion in product design. *TMCE 2010.* Ancona.

Giannini, F., & Monti, M. (2010). A survey of tools for understanding and exploiting the link between shape and emotion in product design. *Proceedings of the TMCE 2010.* Ancona, Italy.

Giannini, F., Montani, E., Monti, M., & Pernot, J.-P. (2011). Semantic Evaluation and Deformation of Curves Based on Aesthetic Criteria. *Computer-Aided Design & Applications, 8*(3), 449-464.

Giannini, F., Monti, M., & Podehl, G. (2004). Styling Properties and Features in Computer Aided Industrial Design. *Computer-Aided Design & Applications, 1*(1-4), 321-330.

Giannini, F., Monti, M., & Podehl, G. (2006). Aesthetic-driven tools for industrial design. *Journal of Engineering Design, 17*(03), 193-215.

Giannini, F., Monti, M., & Podehl, G. (2006). Aesthetic-driven tools for industrial design. *Journal of Engineering Design, 17*(03), 193-215.

Giannini, F., Monti, M., & Podehl, G. (2012). Aesthetic-driven tools for industrial design. *Journal of Engineering Design, 17*(3), 193-215.

Giannini, F., Monti, M., Pelletier, J., & Pernot, J.-P. (2013). A Survey to Evaluate how non Designers Perceive Aesthetic Properties of Styling Features. *Computer-Aided Design & Applications, 10*(1), 129-138.

Godbole, S., & Sarawagi, S. (2004). Discriminative Methods for Multi-labeled Classification. *Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining.*

Graves, A., Fernandez, S., & Schmidhuber, J. (2007). Multi-dimensional Recurrent Neural Networks. *International Conference on Artificial Neural Networks.*

Guillet, S., & Léon, J.-C. (1998). Parametrically deformed free-form surfaces as part of a variational model. *Computer-Aided Design, 30*(8), 621-630.

Gupta, L. D., Malviya, A. K., & Singh, S. (2012). Performance Analysis of Classification Tree Learning Algorithms. *International Journal of Computer Applications, 55*(6), 39-44.

Hall, A. M. (1999). *Correlation-based Feature Selection for Machine Learning.* Hamilton: The University of Waikato.

Hall, A. M., & Smith, A. L. (1999). Feature Selection for Machine Learning: Comparing a Correlation-based Filter Approach to the Wrapper. *Proceedings of the Twelfth International FLAIRS Conference.*

Hand, J. D., & Yu, K. (2001). Idiot's Bayes - Not So Stupid After All? *International Statistical Review, 69*(3), 385-398.

Harada, T., Mori, N., & Sugiyama, K. (1995). Curves' physical characteristics and self-affine properties. *Bulletin of Japonese Society for the Science of Design, 42*(3), 30-40.

Hoffmann, M. C. (1989). *Geometric and Solid Modeling: An Introduction.* San Francisco: Morgan Kauffman.

Hoffmann, M. C. (1993). Implicit curves and surfaces in computer-aided geometric design. *Computer Graphics Application, 13*(1), 79-88.

Hoffmann, M. C., & Joan-Arinyo, R. (1993). Erep: An editable, high-level representation for geometric design and analysis. Dans *Geometric Modeling for Product Realization* (pp. 129-164). North Holland.

Hoffmann, M. C., & Joan-Arinyo, R. (1997). Symbolic Constraints in Constructive Geometric Constraint Solving. *Journal of Symbolic Computation, 23*(2-3), 287-299.

Hoffmann, M. C., & Joan-Arinyo, R. (1998). On user-Defined features. *Computer-Aided Design, 30*(5), 321-332.

Hoffmann, M. C., & Vanecek, G. (1991). Fundamental Techniques for Geometric and Solid Modeling. *Control and Dynamic Systems, 48*, 101-159.

Hou, S., & Ramani, K. (2006). A probability-based unified 3D shape search. *European Commission International Conference on Semantic and Digital Media Technologies, Lecture notes in computer science.*

Hou, S., Lou, K., & Romani, K. (2005). SVM-based semantic clustering and retrieval of a 3D model database. *Journal of Computer Aided Design and Application, 2*(1-4), 124-137.

Hsu, Y.-L. (2010, 11 25). *Optimal Design Laboratory.* Consulté le 04 18, 2015, sur http://designer.mech.yzu.edu.tw/articlesystem/article/compressedfile/(2010-11-25)%20Solid%20modeling%20techniques%20and%20boundary%20representation.pdf

Hullermeier, E., Furnkranz, J., Weiwei, C., & Brinker, K. (2008). Label ranking by learning pairwise preferences. *Artificial Intelligence, 172*(16-17), 1897-1916.

Ip, Y. C., & Regli, C. W. (2005). Content-Based Classification of CAD Models with Supervised Learning. *Computer-Aided Design and Application, 2*(5), 609-617.

Ip, Y. C., Regli, C. W., Sieger, L., & Shokoufandeh, A. (2003). Automated learning of model classification. *Proceedings of the eighth ACM symposium on Solid modeling and applications.*

Jamain, A., & Hand, J. D. (2005). The Naive Bayes Mystery: A classification detective story. *Pattern Recognition Letters, 26*, 1752-1760.

Jeschke, S., Birkholz, H., & Schmann, H. (2003). A Procedural Model for Interactive Animation of Breaking Ocean Waves. *WSCG POSTERS.* Plzen.

Joy, K. (2012). *http://graphics.cs.ucdavis.edu/*. Récupéré sur Institute for Data Analysis and Visualization UC Davis: http://graphics.cs.ucdavis.edu/~joy/GeometricModelingLectures/

Kanaya, I., Nakano, Y., & Sato, K. (2007). Classification of Aesthetic Curves and Surfaces for Industrial Design. *Design Discourse, 2*(4).

Kira, K., & Rendell, A. L. (1992). A Practical Approach to Feature Selection. *In Machine Learning: Proceedings of the Ninth International Conference.*

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence.* Montreal.

Kohavi, R. (1995). *Wrappers for Performance Enhancement and Oblivious Decision Graphs, PhD thesis.* Sandford University.

Kohavi, R., & John, H. G. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*, 273-324.

Kohavi, R., & Sommerfield, D. (1995). Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. *KDD-95.*

Koller, D., & Sahami, M. (1996). Toward Optimal Feature Selection. *Machine Learning: Proceedings of the Thirteenth International Conference.*

Kotthoff, L., Gent, P. I., & Miguel, I. (2010). *An Evaluation of Machine Learning in Algorithm Selection for Search Problems.* London: AI Comunications.

Krakovsky, R., & Forgac, R. (2011). Neural Network Approach to Multidimensional Data Classification via Clustering. *9th International Symposium on Intelligent Systems and Informatics.* Subotica.

La Greca, R. (2005). *Approche déclarative de la modélisation de surfaces, PhD thesis.* Marseille: Université de la Méditerranée .

Laga, H. (2009). 3D Shape Classification and Retrieval Using Heterogenous Features and Supervised Learning. Dans *Machine Learning* (pp. 305-324). InTech.

Lattner, D. A., Miene, A., & Herzog, O. (2004). A Combination of Machine Learning and Image Processing Technologies for the Classification of Image Regions. Dans *Adaptive Multimedia Retrieval* (pp. 185-199). Springer-Verlag Berlin Heidelberg.

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications, 36*, 10896–10904.

Lemm, S., Blankertz, B., Dickhaus, T., & Müller, K.-R. (2011). Introduction to machine learning for brain imaging. *NeuroImage, 56*, 387-399.

Lesot, M.-J., Bouchard, C., Detyniecki, & Omhover, J.-F. (2010). Product shape and emotional design – an application to perfume bottle. *International Conference on Kansei Engineering and Emotion Research.* Paris.

Liu, H., & Yu, L. (2005). Toward Integrating Feature Selection Algorithms for Classification and Clustering. *Knowledge and Data Engineering, 17*(4), 491 - 502.

Lu, X., Suryanarayan, P., Adams, B. R., Li, J., Newman, G. M., & James, W. Z. (2012). On Shape and the Computability of Emotions. *Proceedings of the 20th ACM international conference on Multimedia.* Nara.

Lucas, M., Martin, D., Philippe, M., & Plémenos, D. (1990). Le projet ExploFormes, quelques pas vers la modélisation déclarative de formes. *Bigre n° 67*, 35-49.

Luo, L., & Chen, X. (2013). Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction. *Applied Soft Computing, 13*(2), 806–816.

Maculet, R., & Daniel, M. (2004). Conception, modélisation géométrique et contraintes en CAO: Une synthèse. *Artificial Intelligence Journal, 18*(5-6), 619-645.

Maddouri, M., & Elloumi, M. (2002). A data mining approach based on machine learning techniques to classify biological sequences. *Knowledge-Based Systems, 15*, 217-223.

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Dzeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition, 45*, 3084-3104.

Marhl, M., Guid, N., Oblonšek, Č., & Horvat, M. (1996). Extensions of sweep surface constructions. *Computers & Graphics, 20*(6), 893-903.

Martin, D., & Martin, P. (1988). *An expert system for polyhedral modeling.* Nice: Eurographics.

Meiden, v. d., Hilderick, A., & Bronsvoort, F. W. (2007). Solving topological constraints for declarative families of objects. *Computer-Aided Design*, 652-662.

Michael, N. (2002). *Artificial Intelligence - A Guide to Intelligent Systems.* Essex: Pearson Education Limited.

Mitchell-Guthrie, P. (2014, 08 22). *Subconscious Musings - Advanced analytics from Research Drive to the world*. Récupéré sur SAS: http://blogs.sas.com/content/subconsciousmusings/2014/08/22/looking-backwards-looking-forwards-sas-data-mining-and-machine-learning/

Miura, T. K., & Rudrusamy, U. G. (2014). Aesthetic Curves and Surfaces in Computer Aided Geometric Design. *International Journal of Automation Technology, 8*(3), 304-316.

Miura, T. K., Sone, J., Yamashita, A., & Kaneko, T. (2005). Derivation of a general formula of aestetic curves. *In proceedings of the Eighth International Conference on Humans and Computers .*

Mortenson, E. M. (1995). *Geometric Modeling.* New York: John Wiley and Sons.

Motaal, G. A., El-Gayar, N., & Osman, F. N. (2010). Different Region Identification in Composite Strain-Encoded (C-SENC) Images Using Machine Learning Techniques. Dans *Artificial Neural Networks in Pattern Recognition* (pp. 231-240). Springer-Verlag Berlin Heidelberg.

Muller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. (2006). Procedural modeling of buildings. *ACM Transactions on Graphics*, 614-623.

Nagamachi, M. (2011). *Kansei/Affective Engineering.* Boca Raton, Florida: CRC Press.

Negri, R. G., Dutra, L. V., & Sant-Anna, S. J. (2014). An innovative support vector machine based method for contextual image classification. *ISPRS Journal of Photogrammetry and Remote Sensing, 87*, 241-248.

Ni, L.-P., Ni, Z.-W., & Gao, Y.-Z. (2011). Stock trend prediction based on fractal feature selection and support vector machine. *Expert Systems with Applications, 38*(5), 5569–5576.

Park, S.-H., & Johannes, F. (2007). Efficient pairwise classification. *Proceedings of the 18th European Conference on Machine Learning.*

Patrikalakis, M. N. (2003). *http://ocw.mit.edu/.* Récupéré sur Massachusetts Institute of Technology: http://ocw.mit.edu/courses/mechanical-engineering/2-158j-computational-geometry-spring-2003/lecture-notes/lecnotes1_fixed.pdf

Pernot, J.-P. (2004). *Fully Free Form Deformation Features for Aesthetic and Engineering Design.* PhD Thesis, INP - Grenoble.

Pernot, J.-P., Quao, Q., & Veron, P. (2007). Constraints Automatic Relaxation to Design Products with Fully Free Form Features. Dans *Advances in Integrated Design and Manufacturing in Mechanical Engineering II* (pp. 145-160). Springer.

Petrov, A., Pernot, J.-P., Veron, P., Giannini, F., & Falcidieno, B. (2014). Aesthetic-oriented classification of 2D free-form curves. *TMCE 2014.* Budapest.

Piegl, L. (1991). On NURBS: A survay. *IEEE Computer Graphics and Application, 11*(1), 55-71.

Piegl, L., & Tiller, W. (1997). *The NURBS Book.* Springer-Verlag Berlin Heidelberg.

Plamenos, D. (1994). *La modélisation déclarative en synthèse d'images, tendences et perspectives.* Limoges: MSI 94-03.

Podehl, G. (2002). Terms and Measures for Styling Properties. *International Design Conference - Design 2002.* Dubrovnik.

Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., & Zhang, H.-J. (2009). Two-Dimensional Multi-Label Active Learning with An Efficient Online Adaptation Model for Image Classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence , 31*(10), 1880 - 1897.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann Publishers.

Read, J. (2010). *PhD thesis: Scalable Multi-label Classification.* Hamilton, New Zealand: Department of Computer Science, University of Waikato.

Read, J., Martino, L., & Luengo, D. (2014). Efficient monte carlo methods for multi-dimensional learning with classifier chains. *Pattern Recognition , 47*, 1535-1546.

Read, J., Pfahringer, B., & Holmes, G. F. (2011). Classifier Chains for Multi-label Classification. *Machine Learning, 85*, 333-359.

Ren, Y. (2012). *Design Preference Elicitation, Identification and Estimation, PhD thesis.* The University of Michigan.

Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Englewood: Prentice Hall.

Sajn, L., & Kukar, M. (2011). Image processing and machine learning for fully automated probabilistic evaluation of medical images. *Computer Methods and Programs in Biomedicine, 104*(3), e75-e86.

Schwenker, F., & Trentin, E. (2014). Pattern classification and clustering: A review of partially supervised learning approaches. *Pattern Recognition Letters, 37*(1), 4-14.

Sederberg, W. T., Anderson, C. D., & Goldman, N. R. (1984). Implicit Representation of Parametric Curves and Surfaces. *Computer Vision, Graphics, and Image Processing, 28*, 72-84.

Shhab, A., Guo, G., & Neagu, D. (2001). *A Study on Applications of Machine Learning Techniques in Data Mining.* Bradford: University of Bradford, Department of Computing.

Simon, P. (2013). *Too Big to Ignore: The Business Case for Big Data.* Wiley.

Singh, V., & Gu, N. (2012). Towards an integrated generative design framework. *Design Studies 33*, 185-207.

Smelik, M. R., Tutenel, T., Bidarra, R., & Benes, B. (2014). A Survey on Procedural Modeling for Virtual Worlds. *Computer Graphics Forum, 33*(6), 31-50.

Srinivasana, R., Wang, C., Ho, W. K., & Lim, K. W. (2005). Neural network systems for multi-dimensional temporal pattern classification. *Computers and Chemical Engineering , 29*, 965–981.

Stevens, P. W., Myers, J. G., & Constantine, L. L. (1974). Structured Design. *IBM Systems Journal, 13*(2), 115-139.

Stiny, G., & Gips, J. (1971). Shape Grammars and the Generative Spesification of Painting and Sculptures . *International Federation for Information Processing.*

Su, J., & Zhang, H. (2006). Full Bayesian Network Classifiers. *Proceedings of the 23 rd International Conference on Machine Learning.* Pittsburgh.

Summmers, M. R., & Wang, S. (2012). Machine learning in radiology. *Medical Image Analysis, 16*, 933-951.

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Minnesota: Addison-Wesley.

Tiwari, R., & Singh, P. M. (2010). Correlation-based Attribute Selection using Genetic Algorithm. *International Journal of Computer Applications, 4*(8), 0975 – 8887.

Trohidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. (2008). Multi-Label Classification of Music into Emotions. *The Proceedings of the 9th International Conference on Music Information Retrieval .*

Tsoumakas, G., & Katakis, I. (2007). Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining, 3*(3), 1-13.

Tsoumakas, G., & Vlahavas, I. (2007). Random k-Labelsets: An Ensemble Method for Multilabel Classification. *18th European Conference on Machine Learning.*

Unsalan, C., & Ercil, A. (2001). Conversions between Parametric and Implicit Forms Using Polar/Spherical Coordinate Representations. *Computer Vision and Image Understanding, 81*, 1-25.

Vladimir, V. (1995). *The nature of statistical learning theory.* New York: Springer.

Waikato, U. o. (s.d.). *WEKA*. Consulté le 05 31, 2015, sur http://www.cs.waikato.ac.nz/ml/weka/

Walid, B. A., & Yannou, B. (2007). Unsupervised Beyesian Models to Carry out Perceptual Evaluation in a Design Process. *International Conference en Engineering Design, ICED'07.* Paris.

Wang, X.-W., Nie, D., & Lu, B.-L. (2014). Emotional state classification from EEG data using machine learning approach. *Neurocomputing, 129*, 94-106.

Wasan, S. P., Uttamchandani, M., Moochhala, M. S., & Yap, V. B. (2013). Application of statistics and machine learning for risk stratification of heritable cardiac arrhythmias. *Expert Systems with Applications, 40*, 2476–2486.

Watson, B., Müller, P., Wonka, P., Sexton, C., Veryovka, O., & Fuller, A. (2008). Procedural Urban Modeling in Practice. *IEEE Computer Graphics and Applications* , 18-26.

Witten, H. I., Frank, E., & Hall, A. M. (2011). *Data Mining - Practical Machine Learning Tools and Techniques.* Burlington: Elsevier Inc.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., . . . Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*(1), 37.

Xu, D., & Li, H. (2007). 3D Shape Retrieval Integrated with Classification Information. *Proceedings of the Fourth International Conference on Image and Graphics.*

Xu, K., Kim, G. V., Huang, Q., & Kalogerakis, E. (2015). Data-Driven Shape Analysis and Processing. *Computer Graphics Forum*, 1-27.

Xu, X. (2009). *Integrating Advanced Computer-Aided Design, Manufacturing, and Numerical Control: Principles and Implementations.* IGI Global.

Yoshida, N., & Saito, T. (2007). Quasi-Aesthetic Curves in Rational Cubic Bézier Forms. *Computer-Aided Design & Application, 4*(1-4).

You, H. L., Chang, J., Yang, X., & Zhang, J. J. (2011). Solid modelling based on sixth order partial differential equations. *Computer-Aided Design, 43*, 720-729.

Zaragoza, H. J., Sucar, L. E., Morales, F. E., Bielza, C., & Larranaga, P. (2011). Bayesian Chain Classifiers for Multidimensional Classification. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence.* Barcelona.

Zhang, M.-L., & Zhou, Z.-H. (2006). Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Transactions on Knowledge and Data Engineering , 18*(10), 1338-1351.

Zhang, M.-L., & Zhou, Z.-H. (2007). ML-kNN: a lazy learning approach to multi-label learning. *Pattern Recognition, 40*(7), 2038-2048.

Appendix A:
List of Specifications of the ML algorithms

The following table lists the parameters of the Machine Learning algorithms presented in the thesis. Additionally, specifications of the output model of the selected learning algorithm for the experiments are given in the table.

Name of the algo-rithms	The name of the algorithm in WEKA	Setup parameter of the algorithms (default values)		Specification of the output model – decision tree	
				Classification model CM for Curves	The general classification for Surfaces
C4.5	Decision tree J48	ConfidenceFactor: minNumObj: numFolds: reducedErrorPruning: saveInstanceData: seed: unpruned: subtreeRaising: useLaplace: binarySplits:	0.25 2 3 false false 1 false true false false	Number of leaves: 249 Number of nodes: 248 Size of the tree: 497	Number of leaves: 135 Number of nodes: 134 Size of the tree: 269
K – Nearest Neighbors (KNN)	k Insance-Based classifier IBk	KNN: crossValidate: distanceWeighting: meanSquared: SearchAlgorithm:	10 false no weighting false EuclidiDistance		
Support Vector Machine (SVM)	Sequential Mi-nimal Optimiza-tion SMO	filterType: kernel: numFold: epsilon: randomSeed: toleranceParameter: checkTurnedOff: c buildLogisticModels	Normalization PolyKernel -1 1.0E-12 1 0.001 false 1.0 false		
Naïve Bayes (NB)	NaiveBayes	ModelInOldFormat: useKernelEstimator: SupervisedDiscretization:	false false false		
RIPPER	Decision rules JRip	checkErrorRate: folds: minNo: optimizations: seed usePruning:	true 3 2.0 2 1 true		

Appendix B:
List of Abbreviations

The following table describes the signification of various abbreviations and acronyms used throughout the thesis. Standard and Nonstandard acronyms that are used in some places to abbreviate the names of certain white matter structure are given in this list.

Abbreviation	Meaning
DM	Data Mining
MLT	Machine Learning Techniques
AE	Affective Engineering
AM	Additive Manufacturing
CAD	Computer Aided Design
CNC	Computer Numerical Control
DM	Declarative Modeling
PDP	Product Design Process
NURBS	Non-Uniform Rational B-Spline
FIORES	Formalization and Integration of an Optimized Reverse Engineering Styling Workflow
IDS	Initial Data Set
GUI	Graphical User Interface
AI	Artificial Intelligence
GM	Geometric Modeling
CSG	Constructive Solid Geometry
B-rep	Boundary Representation
SG	Shape Grammars
GD	Generative Design
LS	L-systems
CA	Cellular Automata
GA	Generic Algorithms
SI	Swarm Intelligence
CAS	Computer-Aided Styling
FFF	Free-Form Feature
LDDC	Logarithmic Distribution Diagram of Curvature
LA	Log-Aesthetic
LDDC	Logarithmic Distribution Diagram of Curvature
LCG	Logarithmic-Curvature Graph
AC	Aesthetic Curves
GCS	Generalized Cornu Spiral
GLAC	Generalized Log-Aesthetic Curves
KDD	Knowledge Discovering from Data
WEKA	Waikato Environment for Knowledge Analysis
CLI	Command Line Interface
SVM	Support Vector Machine
SMO	Sequential Minimal Optimization
kNN	k – Nearest Neighbors
CART	Classification And Regression Trees
IBk	k – Instance Based classifier

RIPPER	Repeated Incremental Pruning to Produce Error Reduction
NN	Neural Networks
NB	Naïve Bayes
DT	Decision Trees
DR	Decision Rules
ANN	Artificial Neural Networks
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
FS	Feature Selection
CFS	Correlation-based Feature Selection
MLC	Multi-Label Classification
BR	Binary Relevance
PW	Pairwise Classification
LC	Label Combination
RT	Ranking and Threshold
MDC	Multi-Dimensional Classification
BN	Bayesian Networks
IS	Independent Classifiers
CC	Classifier Chains
MCC	Monte-Carlo Classifier Chains
BCC	Bayesian Chain Classifiers
FE	Finite Element
IGES	Initial Graphics Exchange Specification
TRS	Training Set
TES	Testing Set
CM	Classification Model
CMr	Classification Model created using dimensionless attributes
AS	Attribute Selection
S	Straightness
F	Flat
AF	Almost Flat
NF	Not Flat
VNF	Very Not Flat
ARFF	Attribute-Relation File Format
Ts	Target shapes
TsCM	Target shapes of a Coffee Machine
TsCB	Target shapes of a Car Back
TsCD	Target shapes of a Car Door
DP	Deformation Path
TS	Target Surfaces
M	Morphing
PCA	Principal Component Analysis
MBB	Minimal Bounding Box

Content

UNDERSTANDING THE RELATIONSHIPS BETWEEN AESTHETIC PROPERTIES OF SHAPES AND GEOMETRIC QUANTITIES OF FREE-FORM CURVES AND SURFACES USING MACHINE LEARNING TECHNIQUES

ABSTRACT : Today on the market we can find a large variety of different products and different shapes of the same product and this great choice overwhelms the customers. In such situations, how to make a choice? Which product we would buy? It is evident that the aesthetic appearance of the product shape and its emotional affection will lead the customers to the decision for buying the product. Designing appealing objects plays a key role in the commercial success of a product. Therefore, it is very important to understand the aesthetic proper-ties and to adopt them in the early product design phases. Additionally, being able to design shapes by manipulating their aesthetic properties while controlling the evoked positive emo-tions, will help in faster reaching the customers on the market. The objective of this thesis is to propose a generic framework for mapping aesthetic properties to 3D free-form shapes, so as to be able to extract aesthetic classification rules and associated geometric properties. The extraction of the aesthetic classification rules is based on considering the perception of the aesthetic properties of non-professional designers (potential customers) by conducting interviews. The key element of the proposed framework is the application of the Data Min-ing (DM) methodology and Machine Learning Techniques (MLTs) in the mapping of aesthetic properties to the shapes. The overall framework describes the main activities of how to in-vestigate the mapping of aesthetic properties to free-form shapes and by modifying any of its constitutive elements (i.e. base, beam, pillars and roof) to change the final results (i.e. the configured classifier). First, this framework has been validated on curves and its aesthetic property, i.e. straightness, and then extended to free-form surfaces and its aesthetic proper-ty – Flatness. The application of the framework is to investigate whether there is a common judgment for the flatness perceived from non-professional designers. Despite this, the pro-posed framework also investigates two additional aspects: how the size of the surrounding and the transition towards the surrounding affects the perception of the flatness.

The aim of the framework is not only to establish a structure for mapping aesthetic properties to free-form shapes, but also to be used as a guided path for identifying a map-ping between different semantics and free-form shapes. The integration of the aesthetic aspects in the industrial design process contributes in developing of the declarative modeling approach. The aim of the integration is to open new perspectives for developing of user-friendly manipulation tools for modifying the free-form shapes. The long-term objective of this work is to define a methodology to efficiently integrate the concept of Affective Engi-neering in the Industrial Designing.

ARTS ET MÉTIERS ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

EXPLOITATION DE TECHNIQUES D'APPRENTISSAGE ARTIFICIEL POUR LA COMPREHENSION DES LIENS ENTRE LES PROPRIETES ESTHETIQUES DES FORMES ET LES GRANDEURS GEOMETRIQUES DE COURBES ET SURFACE GAUCHES

Résumé : Aujourd'hui, sur le marché, on peut trouver une vaste gamme de produits différents ou des formes variées d'un même produit et ce grand assortiment fatigue les clients. Dans une telle situation, comment faire le choix ? Quel produit acheter ? Il est clair que la décision des clients d'acheter un produit dépend de l'aspect esthétique de la forme du produit et de l'affection émotionnelle. Concevoir des objets attirants joue un rôle essentiel dans le succès commercial d'un produit. Par conséquent, il est très important de comprendre les propriétés esthétiques et de les adopter dans la conception du produit, dès le début. En outre, être en mesure de concevoir des formes en utilisant leurs propriétés esthétiques tout en contrôlant les émotions positives évoquées aide à accéder plus rapidement vers les clients sur le marché. L'objectif de cette thèse est de proposer un cadre générique pour la cartographie des propriétés esthétiques des formes gauches en 3D en façon d'être en mesure d'extraire des règles de classification esthétiques et des propriétés géométriques associées. L'extraction des règles de classification esthétique repose sur l'étude de la perception des propriétés esthétiques des concepteurs non-professionnels (clients potentiels) en menant des interviews. L'élément clé du cadre proposé est l'application des méthodologies de l'Exploration des données (Data Mining) et des Techniques d'apprentissage automatiques (Machine Learning Techniques) dans la cartographie des propriétés esthétiques des formes. Le cadre général décrit les activités principales de la façon d'enquêter la cartographie des propriétés esthétiques des formes gauches et en modifiant l'un de ses éléments constitutifs (c'est-à-dire, la base, la poutre, les piliers et le toit) modifier les résultats finaux (c'est-à-dire, le classificateur configuré). Premièrement, ce cadre a été validé sur des courbes et sa propriété esthétique, à savoir la rectitude, ensuite, étendu aux surfaces de la forme libre et sa propriété esthétique - planéité. L'application du cadre est d'étudier s'il y a une opinion commune pour la planéité perçu de la part des concepteurs non-professionnels. À part cela, le cadre proposé étudie également deux autres aspects: comment la taille de l'environs et la transition vers l'entourage affectent la perception de la planéité.

Le but de ce cadre n'est pas seulement d'établir une structure pour repérer des propriétés esthétiques des formes gauches, mais aussi pour être utilisé comme un chemin guidé pour l'identification d'une cartographie entre les sémantiques et les formes gauches différentes. L'intégration des aspects esthétiques dans le processus du design industriel contribue au développement de l'approche de la modélisation déclarative. Le but de l'intégration est d'ouvrir de nouvelles perspectives pour le développement d'outils de manipulation pratiques pour modifier les formes gauches. L'objectif à long terme de ce travail est de définir une méthodologie pour intégrer efficacement le concept de l'Ingénierie affective (c.à.d. Affective Engineering) dans le design industriel.

Mots clés : Courbes et surfaces gauches, modélisation géométrique, modélisation déclarative, extraction de données et techniques d'apprentissage automatiques, propriétés esthétiques, ingénierie affective, design industriel, fabrication additive

LA COMPRENSIONE DELLE RELAZIONI TRA LE PROPRIETA ESTETICHE DELLE FORME E GRANDEZZE GEOMETRICHE DELLE CURVE E SUPERFICI A FORMA LIBERA UTILIZZANDO TECNICHE DI APPRENDIMENTO AUTOMATICO

Sommario : Oggi sui mercati si trova una grande varietà di prodotti ed una grande varietà di forme dello stesso prodotto e questa scelta fa' confondere i consumatori. In una situazione come questa come fare la scelta? Quale prodotto vorremmo comprare? E' evidente che l'aspetto estetico di un prodotto ed il suo effetto emotivo portano i consumatori alla decisione di comprare un prodotto. La progettazione di oggetti attraenti gioca il ruolo chiave per il successo commerciale del prodotto, dunque è molto importante capire le caratteristiche estetiche ed adottarle nelle prime fase della progettazione del prodotto. Inoltre, essendo in grado di disegnare le forme manipolando le loro caratteristiche estetiche, mentre si controllano le emozioni positive evocate, aiuterà a raggiungere i clienti sul mercato più velocemente.Lo scopo di questa tesi è quello di proporre un quadro generico per la mappatura delle proprietà estetiche alle forme libere 3D per essere in grado di estrarre regole di classificazione estetica e relative proprietà geometriche. L'estrazione delle regole di classificazione estetica si basa sulla percezione delle proprietà estetiche dei designer non professionisti (potenziali clienti) conducendo interviste. L'elemento chiave del quadro proposto è l'applicazione del Data Mining (DM) metodologie e tecniche di apprendimento automatico (MLT) nella mappatura delle proprietà estetiche alle forme Il quadro generale descrive le attività principali di come studiare la mappatura delle proprietà astetiche sulle free-form forme e modificando uno dei suoi elementi costitutivi (base, fascio, pilastri e tetto)di modificare i risultati finali (cioè classificatore configurato). Inizialmente questo quadro è stato convalidato su curve e le sue proprietà estetiche ovvero rettilineità, e poi sulle free-form superfici e le sue proprietà estetiche – flatness. L'applicazione del quadro è indagare se esiste un giudizio comune per la planarità (flatness) percepita da designer non professionisti. Nonostante ciò, il quadro proposto indaga anche altri due aspetti: come la dimensione della zona circondante e la transizione verso la zona circondante colpisce la percezione della planarità (flatness).

Lo scope del quadro non è solo quello di stabilire una struttura per mapping (mappare) le proprietà estetiche per free-form forme, ma anche per essere usato come una guida per l'identificazione di una mappatura tra diverse semantiche e free form forme. L'integrazione degli aspetti estetici nel processo di design industriale contribuisce allo sviluppo di un approccio di modellazione dichiarativo. L'obiettivo dell'integrazione è quello di aprire nuove prospettive per lo sviluppo di strumenti di manipolazione di utilizzo facile per la modifica delle free form. L'obiettivo di questo lavoro a lungo termine è quello di definire una metodologia per integrare il concetto di Affective Engeneering nel Industrial Designing in modo efficace.

Parole Chiave : Curve e superfici a forma libera, la modellazione geometrica, modellazione dichiarativa, data mining e tecniche di machine learning, proprietà estetiche, ingegneria affettivo, design industriale, produzioni additive

EXPLOITATION DE TECHNIQUES D'APPRENTISSAGE ARTIFICIEL POUR LA COMPREHENSION DES LIENS ENTRE LES PROPRIETES ESTHETIQUES DES FORMES ET LES GRANDEURS GEOMETRIQUES DE COURBES ET SURFACE GAUCHES

RESUME: Aujourd'hui, sur le marché, on peut trouver une vaste gamme de produits différents ou des formes variées d'un même produit et ce grand assortiment fatigue les clients. Il est clair que la décision des clients d'acheter un produit dépend de l'aspect esthétique de la forme du produit et de l'affection émotionnelle. Par conséquent, il est très important de comprendre les propriétés esthétiques et de les adopter dans la conception du produit, dès le début. L'objectif de cette thèse est de proposer un cadre générique pour la cartographie des propriétés esthétiques des formes gauches en 3D en façon d'être en mesure d'extraire des règles de classification esthétiques et des propriétés géométriques associées. L'élément clé du cadre proposé est l'application des méthodologies de l'Exploration des données (Data Mining) et des Techniques d'apprentissage automatiques (Machine Learning Techniques) dans la cartographie des propriétés esthétiques des formes. L'application du cadre est d'étudier s'il y a une opinion commune pour la planéité perçu de la part des concepteurs non-professionnels. Le but de ce cadre n'est pas seulement d'établir une structure pour repérer des propriétés esthétiques des formes gauches, mais aussi pour être utilisé comme un chemin guidé pour l'identification d'une cartographie entre les sémantiques et les formes gauches différentes. L'objectif à long terme de ce travail est de définir une méthodologie pour intégrer efficacement le concept de l'Ingénierie affective (c.à.d. Affective Engineering) dans le design industriel.

Mots clés : Courbes et surfaces gauches, techniques d'apprentissage automatiques, propriétés esthétiques, ingénierie affective, design industriel.

UNDERSTANDING THE RELATIONSHIPS BETWEEN AESTHETIC PROPERTIES OF SHAPES AND GEOMETRIC QUANTITIES OF FREE-FORM CURVES AND SURFACES USING MACHINE LEARNING TECHNIQUES

ABSTRACT: Today on the market we can find a large variety of different products and different shapes of the same product and this great choice overwhelms the customers. It is evident that the aesthetic appearance of the product shape and its emotional affection will lead the customers to the decision for buying the product. Therefore, it is very important to understand the aesthetic proper-ties and to adopt them in the early product design phases. The objective of this thesis is to propose a generic framework for mapping aesthetic properties to 3D free-form shapes, so as to be able to extract aesthetic classification rules and associated geometric properties. The key element of the proposed framework is the application of the Data Mining (DM) methodology and Machine Learning Techniques (MLTs) in the mapping of aesthetic properties to the shapes. The application of the framework is to investigate whether there is a common judgment for the flatness perceived from non-professional designers. The aim of the framework is not only to establish a structure for mapping aesthetic properties to free-form shapes, but also to be used as a guided path for identifying a mapping between different semantics and free-form shapes. The long-term objective of this work is to define a methodology to efficiently integrate the concept of Affective Engineering in the Industrial Designing.

Keywords : Free-form curves and surfaces, machine learning techniques, aesthetic properties, affective engineering, industrial design.