HAL
open science

Processing free form objects within a Product Development Process framework

Bianca Falcidieno, Franca Giannini, Jean-Claude Léon, Jean-Philippe Pernot

To cite this version:

Bianca Falcidieno, Franca Giannini, Jean-Claude Léon, Jean-Philippe Pernot. Processing free form objects within a Product Development Process framework. Advances in Computers and Information in Engineering Research, 1, ASME-Press, pp.317-344, 2014, 10.1115/1.860328_ch13 . hal-00922950v2

HAL Id: hal-00922950

https://hal.science/hal-00922950v2

Submitted on 25 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contents

1 Processing free form objects within a Product Development Process framework **3**
Bianca Falcidieno, Franca Giannini, Jean-Claude Léon and Jean-Philippe Pernot
1.1 Introduction 3
1.2 Product Development Process and free-form surfaces in product aesthetic and engineering designs 5
1.3 Modeling and sketching free-form surfaces 11
1.4 Toward advanced models, methods, and tools for modeling free-form surfaces . . 20

1

Processing free form objects within a Product Development Process framework

Bianca Falcidieno

IMATI-GE CNR, Genova, Italy

Franca Giannini

IMATI-GE CNR, Genova, Italy

Jean-Claude Léon

Grenoble University - INRIA, Grenoble, France

Jean-Philippe Pernot

Arts et Métiers Paris Tech, Aix-en-Provence, France

1.1 Introduction 3
1.2 Product Development Process and free-form surfaces in product aesthetic and engineering designs 5
1.3 Modeling and sketching free-form surfaces 11
1.4 Toward advanced models, methods, and tools for modeling free-form surfaces 20
References 26

1.1 Introduction

Creativity is the generation of new ideas, by proposing new ways of looking at existing problems, seeing new opportunities, or by exploiting emerging technologies or changes in markets. Innovation is the successful exploitation of new ideas. Design is what links creativity and innovation.

Creative industries are based on these three processes. Combined with information and communication technologies (ICT), they are an emerging lead market in the world knowledge economy and a dynamic motor for economic and social innovation, outlining a common vision of future ICT-driven creative industries, where the fusion of creativity and technology is considered a crucial factor of success.

We are observing a huge impact of ICT on several sectors of these industries: music composition and production; film, television, and video; animation and computer games; writing, publishing, and print media; advertising and marketing; architecture and visual arts; product design and manufacturing. Activities in all these sectors require a high level of technical knowledge to create and manipulate content. The computer is now the workbench for making digital content, and creators need to feel at home with digital technology.

In most of these contexts the advances in technology pave the way for new ICT services, where dynamically evolving applications are continuously adapting to changing user needs and to user-generated content. This paradigm is supporting inventive and creative practices in the arts, science,

engineering, education, and business by enabling entirely new types of creative digital productions. It pushes advances in user-centric content creation systems and the convergence of roles of professional and non-professional users, who become the major content producers, whose focus is not only on the technologies but, even more, on the applications where users are working.

Recently, the content sector has experienced a massive transformation due to changes in user behavior, showing a growing desire to participate actively in the creation process, stimulated by new ICT capabilities. This is particularly true in fields like entertainment, digital media, and visual arts, where a product will be judged as creative to the extent that it is both a novel and appropriate response to the task in hand, and the task is heuristic rather than algorithmic [7]. Also in product design, new technologies such as low-cost scanners or rapid prototyping technologies give the impression that anybody can become a creative person, thereby creating his/her own objects.

However, besides those trends promising prosperous connections between creative sectors and ICT tools, several obstacles prevent this vision from turning into reality. Designing is a cyclic, iterative loop of phases or steps, a form of systematic problem solving, finding a solution and integrating the procedural aspects of design with the structural and feasibility aspects of design problems. Design is also a heuristic interactive process that can be adapted to the particular requirements of a task [19]. It is therefore crucial that computer-aided tools adopted during industrial product development can support users in dealing with these paradigms. In particular, designing industrial products to capture and render emotion and intuition in the aesthetic of shapes is crucial for the success to the market.

Industrial design includes the specification of all those products that are somehow subject to visual judgment and appreciation. It ranges from transport vehicles, not only cars but also boats, buses, trains, to home appliances, furniture, cosmetic containers, and thousands of other objects that we deal with in our daily life. Therefore, it is not surprising that the industrial design process is not unique and may vary deeply from company to company, depending on several factors such as the complexity of the product to be designed, the available financial resources, the equipment used, the team of specialists involved and so on. For simple manufactured objects a single designer can represent the design team, but for complex products a team of specialists is usually required. In the latter case, interaction with the engineering team, which is in charge of guaranteeing the fulfillment of planned functionality and performances together with product feasibility, is much more complex. It is clear that the more technologically complex a product, the more problems will arise when the stylists idea is coupled with engineering and ergonomic constraints. Existing interfaces and tools, though functional and very powerful, appear cumbersome and daunting to a stylist; creating a complex model still requires considerable technological knowledge and a useless effort in specifying unknown details, which are not necessary in the early phases of design and limit freedom and freshness in conception.

To a larger extent, this is true when designing objects using free-form surfaces, where other functional issues rather than aesthetics are prominent, and it is impossible to precisely specify a shape at the beginning due to both an uncertainty of the idea itself (not fully refined at the conceptual phase) and the lack of information about engineering conditions and simulation results. The consequence is that the design process requires too many back-and-forth attempts before obtaining acceptable outcomes, thus being still far from the stylist's needs of intuitive systems that improve interactivity without reducing their creativity.

The purpose of the chapter is then to review: (1) the stages of a product development process (PDP) where free-form shapes appear and are subjected to aesthetic and functional requirements; (2) the modeling, sketching, and modification activities illustrating how free-form surfaces can be processed and what are the corresponding difficulties faced during these tasks; (3) the contributions of our community to solve some of these issues, and the problems which are still open.

The complete product definition can be seen as the result of process where all the various requirements (functional, aesthetic, economical, feasibility, and so on) are satisfied at the best. For this reason PDPs involve many actors, tasks, and activities, currently supported by dedicated IT tools

such as: computer-aided design (CAD), computer-aided styling (CAS), computer-aided engineering (CAE), and computer-aided manufacturing (CAM). These processes often rely on digital mock-ups (DMUs) integrating multi-representations and multi-resolution geometric models to shape complex components and products containing notably the needed free-form surfaces. Free-form surfaces and free-form shapes appeared long ago in CAD/CAS systems based on fairly sophisticated modeling functions. Little progress has been obtained since then because free-form shapes are still obtained tediously even if reverse engineering (RE) processes have become rather common practices. There are still several challenging limitations and difficulties that need to be addressed to improve the complete definition of products incorporating free-form shapes, ranging from the initial model creation to its final assessment.

Problems and solutions related to this topic are presented in the rest of the chapter: Section 1.2 deals with PDP and free-form surfaces in product aesthetic and engineering designs; Section 1.3 focuses on modeling and sketching free-form surfaces; Section 1.4 explores advanced models, methods, and tools for modeling and processing free-form surfaces. The last section outlines conclusive remarks and open issues.

1.2 Product Development Process and free-form surfaces in product aesthetic and engineering designs

As stated in the previous section, the product design process and, more generally, PDPs, are still activities where new tools and new approaches can support creation and innovation. The purpose of this section is to focus on design processes, as part of PDPs, and their interactions with other processes as a framework to identify some characteristic configurations involving free-form surfaces processing that significantly reduce the efficiency of creation and design activities and, consequently, that of PDPs. First, a generic structure of a PDP is proposed as a reference process that is a common framework to derive the contexts where difficulties in free-form shape processing will be illustrated.

1.2.1 PDP structure

PDPs, as experienced by companies, do not have a unique structure but vary in accordance with the products they deliver, their configuration, and their complexity. The development of domestic appliances significantly differs from that of an aircraft. Clearly, the complexity of the latter, i.e., its number of components, its number of functions, the diversity of physical phenomena it is subjected to, is much greater than that of the former thus requiring, for instance, a larger number of engineering verifications and simulations. The PDP structure as well as the product complexity can strongly influence the activity of designers and/or engineers during free-form surface processing whenever this category of surface is relevant for a given product. However, a generic structure of a PDP can be derived as a reference scheme that can be adapted to specific companies and classes of products. It is assumed that a digital model of the product stands as a reference product model, e.g., most of the time CAD models are regarded as product reference data. Also, prototypes, if they are part of a PDP, are highlighted here for simplicity. Figure 1.1 summarizes the structure of such a reference PDP. In it, arrows represent the flow of information and/or digital models that can be communicated from one activity to another as long as the first activity is carried out. Double-headed arrows are not prescribing systematic communications between two or more activities. They can reduce to single-way communications for some specific PDPs and it is the purpose of this section to further explain when and why this can happen.

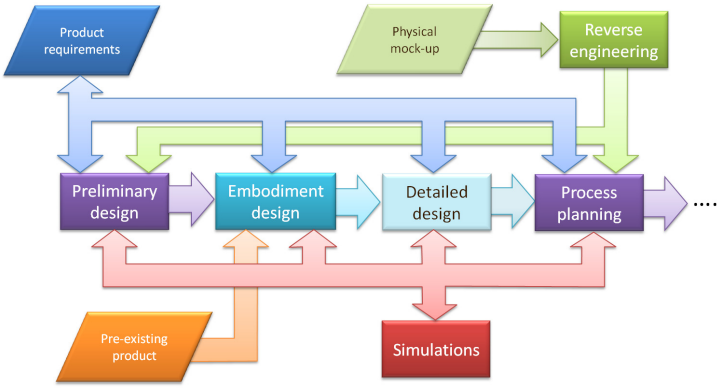


FIGURE 1.1: Reference structure of a PDP highlighting its major processes that relate to product design and involve free-form shapes.

The proposed task decomposition of this generic PDP has to be tuned for each component of a product, i.e., the reverse engineering task may be used or applicable to a few components only or not at all. Similarly, the re-engineering task, if applicable, may be performed using a preexisting model for a specific product's components or sub-assemblies. Also, tasks like preliminary design may nearly vanish when products are simple. Structuring a PDP is more and more critical as product complexity increases. Often, better and more innovative products require incorporating more functions, e.g., the evolution from initial cell phones to smart phones, hence increasing their number of components. They also require a more accurate evaluation of their behavior and performances in possibly new usage contexts, thus the compromise among the various requirements and the use of optimization techniques are becoming increasingly frequent. Consequently, behavior optimization, e.g., structural behavior using finite element (FE) models, covers shape optimization processes, which refine shapes, increasing their complexity in the sense of number of modeling features required to describe them. Any optimization process calls for more structure between either of the design and simulation tasks in a PDP to tightly connect simulation and design updates, i.e., a CAD model and a FE model, when component shapes are evolving. This is illustrated in Figure 1.1 with double-headed arrows between design processes and simulation to enforce that CAD and FE models must interact with each other during a shape optimization process. Design models are not only feeding simulations but simulation results can act on these models. Such connections between digital models increase the structure of a PDP.

The task breakdown used here is classical and it is compatible with concurrent engineering approaches when considering that components are not uniformly moving from preliminary design to embodiment design and that some analysis can be performed simultaneously. This means that some sub-assemblies or components of a product are not at the same design stage as others at a given time of a PDP. For example, a car dashboard can be at the embodiment design stage whereas the internal areas of a car body trunk, defined by a set of sheet metal components, can be at the level of a detailed design.

The reference structure of PDPs incorporates the necessary source of input data, i.e., the product requirements, and an important process, i.e., the simulation that can take place at different levels of PDPs. The product requirements refer to the functional and aesthetic specifications leading the design choices and the criteria for the evaluation and selection of the design option that performs best when the product is in use. As illustrated in Figure 1.1, requirements can influence all the stages of a PDP from preliminary design to process planning. Also, PDP stages antecedent to a given one can set requirements on the subsequent stages. Considering the example of a car body

design, the outer car body shape is defined at the preliminary design stage and sets dimensional requirements, e.g., doors and headlights, to the inner body structure that will be designed at later stages of a PDP. Similarly, the simulation process designates a generic activity that takes place all along a PDP and refers to different categories of simulations. Classically, they refer to behavioral evaluation of components. Digital behaviors have been extensively developed and now cover a wide range of configurations and phenomena. Classically, simulations address physical phenomena, i.e., structural behavior, acoustics, thermal performance, fluid flow. Nowadays, this is evolving with user-centered simulations, i.e., using virtual reality environments and tools, showing that simulation diversity as well as its number are still increasing in order to reduce the number of prototypes and speed up PDPs. Also, the 3-D printing process can be regarded as a particular type of simulation where the objectives can be to assess qualitatively the visual effect of free-form shapes, their size in a real environment, their color, and so on. The higher the complexity of a product, the greater the number of simulations tend to be to obtain a designed product satisfying the desired requirements.

Simulations can interact both ways with the different stages of design. They are not only processes analyzing the geometric models resulting from the design activities but they may provide updated components' or assemblies' geometry. This happens when the simulation turns into an optimization process acting on the 3-D shape's design parameters.

Focusing on products highly characterized by free-form surfaces, we can further identify some peculiarities in the PDP organization.

Even if we assume that the reference model is a digital one, it must be noted that such a digital model does exist for all the PDP stages in all the contexts. For instance, at the preliminary design phase not all the companies are directly using digital tools, even if this is the trend. As an example, the preliminary design of a car body is generally still based on hand sketches. The digital model comes later on in the PDP at a stage that can be rooted at the end of the preliminary design stage. This is due to the precision still needed by computer-aided tools inappropriate to let designers express their creativity with user-friendly modeling systems.

The "pre-existing" product item in Figure 1.1 expresses configurations often designated as product re-engineering, where a company reconsiders existing design and manufacture solutions. There, assuming that the pre-existing product owns a digital representation, the modification processes are central to the re-engineering task. To this aim, compatibility among legacy systems and versions in data management is crucial. Unfortunately, this is not guaranteed over long term periods. Consequently, model parameterization and construction trees are frequently not available. In addition, in the case of free-form shapes, history trees are frequently incomplete and contain useless elements, thus making model adaptation a difficult and time-consuming task. Having the ability to modify easily the pre-existing digital models, and particularly free-form surfaces, is a recurrent requirement that will be appearing in other configurations. Reverse engineering is a typical process frequently carried out for free-form shaped components every time an existing component does not have a digital counterpart or when a physical mock-up, i.e., a product prototype, is created for some evaluation purposes and modified. For instance, in the automotive field, clay models are sometimes created before the mass production to evaluate the car at the real size, and are the origin of a reverse engineering phase leading to the digital model of the detailed design phase. Moreover, 3-D printing is becoming more and more frequent as a tool for evaluating design solutions in other fields.

The previous considerations already show some alternatives influencing PDP pipelines, showing how PDP pipelines can be derived from the reference one proposed in Figure 1.1 to match practices in different companies.

Overall, a PDP cannot be reduced to a simple process flow extracted from Figure 1.1. Even if a main objective is to get a product right at the first time, addressing one or several PDP processes at a finer resolution, some requirements may not be achieved correctly with the first design. Consequently, shape modifications and dimension adjustments are common practices that must be supported by the digital models of products and their components. These notions of modifications and adjustments are tightly related to the concept of compromise that is taking place all along the

various processes of a PDP.

Another important aspect characterizing the PDP process is the number of software systems involved and the types of digital parts descriptions on which they act. If B-Rep CAD models are commonly regarded as product reference data, other digital shape representations are anyhow processed, e.g., FE or finite volume models for simulations. Other digital shape models like point sets, images, take part to reverse engineering processes. Moreover, in some specific cases, additional shape-related data may be available in some form. This is the case, for instance, of medical prosthesis design where data acquired from the patient are available and constitute design constraints. Another case is the maintenance of Peltons water turbines, where the model of the product and digital acquisitions of the real one are combined together to evaluate the water abrasive effects. Then, this comparison is used to determine the amount of material to add to recover the initial performance of the turbine. This diversity of digital representations somehow characterizes the PDP tailored to a given product, which enforces the diversity of scenarios that can be derived from the reference process structure in Figure 1.1. In any case, dedicated adaptation processes have to be developed to answer the needs for multi-representations and multi-resolutions at the different stages of a PDP. Such adaptations also have to take into account interoperability conditions that are key issues when setting up a PDP within a competitive industrial environment.

The various PDP processes generate or modify not only shape models but also engineering data, e.g., component names, material designation, and mechanical properties, stress fields, functions, and so on. Some of these data can be tightly related to shape models, i.e., to geometric information. Here, considering geometric information as a reference, these data are collectively designated as semantic information. Most often, semantic and geometric data are loosely connected and their connection is not robust, i.e., semantic data are assigned interactively to a whole component rather than to the appropriate geometric entities or to some of its sub-domains but it is error prone and tedious. Producing and maintaining tighter connections between them is also a key to an improved efficiency of PDP processes [48]. This can also be a means to improve the level of interoperability between digital models and processes of a PDP.

1.2.2 Free-form shape and PDP relationships

This section focuses on the digital representations of a component or a product to highlight some key issues. Digital models exist at most stages of a PDP. The preliminary design phase can be considered as the only one where digital models may not be available.

As mentioned in the previous section, hand-sketched drawings are frequently used to define free-form shapes such as the external shape of a car or the shape of home appliances. Computer-aided styling (CAS) tools and CAD ones are still far from supporting the designers' creativity with appropriate user interfaces and concepts. Figure 1.2 illustrates to which extent hand-sketched documents contain uncertainties, showing that they cannot be interpreted as sharp product requirements and can hardly be computer processed to produce a 3-D model directly from sketches [43, 49]. More precisely, projection lines conveyed by feature points are not converging consistently toward a vanishing point in the perspective view of Figure 1.2(a) and vanishing points that contribute to the perspective views of Figures 1.2(a) and 1.2(b) cannot be located accurately.

Similar observations can be done about functional requirements, where specifications are not always consistent, for example, dimensional values may be not compatible with material strength prescriptions.

The above configurations reflect the need for compromises (see Section 1.2) inside or between PDP processes and these compromises take place throughout a PDP. Product requirements can be subdivided into two complementary categories, qualitative and quantitative ones. Quantitative requirements like a power, a velocity, and so on, can be subjected to tolerances, which express the compromise. Qualitative requirements like aesthetics cannot be associated with tolerances and compromises are more subjective. To express the concept of compromise at the level of digital shape

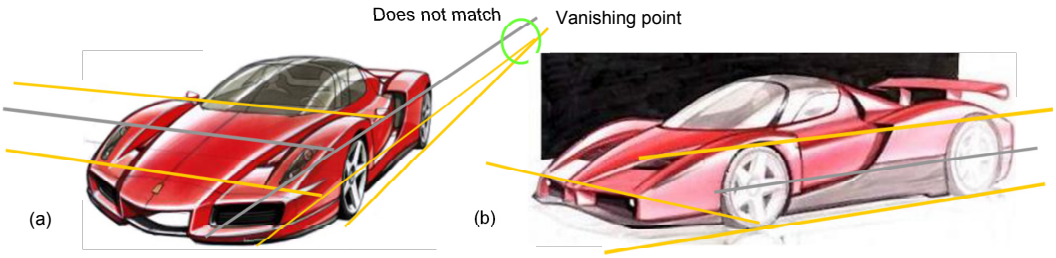


FIGURE 1.2: An example configuration where sketched shapes are not accurate enough to be used as geometric prescription. (a) An analysis of a sketch showing that lines converging toward a focal point are inconsistent. Other lines cannot define precisely the focal point because they are almost parallel. (b) lines can hardly define the focal point showing that the sketch can contain inaccuracies (courtesy of Pininfarina).

models, it is a means to set requirements on these digital representations. Their capabilities to be easily modified, i.e., add/remove shape details, modify dimensions, or apply geometric constraints, are among these requirements. Because modifying a shape is also comparable to exploring a solution domain, letting the user know that some shape modifications are just not possible is as important as providing solutions. As an example, generating a fillet of arbitrary radius between two free-form surfaces has no systematic solution. When seeking a compromise during a PDP process, computing the maximal radius compatible with a configuration of free-form surfaces is as important as looking for the maximal mechanical stresses in a component.

It can be observed that free-form surface models do not benefit from construction tree structures that ease modifications and associate a process with a digital representation of an object (see Figure 1.9(b)). Classically, industrial CAD software rely on an incremental modeling paradigm where a complex shape is generated through a sequence of simple shape modeling operations (see Section 1.3.4). Commonly, the chronology of these operations is at the basis of a history tree describing the construction process of an object.

Consequently, free-form shape modifications are generally tedious because operator parameterization and construction tree are often missing, or partly missing, and this combines with frequent update failures because free-form surfaces may not exist. It is always the case when geometric models are transferred between different software systems as it is often required to carry out the various PDP activities (see Figure 1.1). Indeed, during a model transfer between CAD and CAE software, whatever the standard format used, i.e., STEP or IGES, the construction tree of an object is not transferred. Hence shape modifications become tedious anyway. To overcome this difficulty, shape modifications based on deformation processes is an approach that has been extensively addressed (see Section 1.3). However, defining a structure that can be an intrinsic history representation of a free-form surface generation is still a means for significantly improving its generation, modification, and shape optimization.

Also, mechanical parts may be characterized by free-form surfaces directly subjected to functional requirements. A characteristic configuration, where functional requirements are key issues and can be exemplified, is the definition of turboreactor blades (see Figure 1.3). Such components are characterized by canonical surfaces forming a subset of their shape and free-form ones defining their aerodynamic area, at least. These free-form surfaces are commonly generated by interpolating sections defining blade profiles. This generative process barely contains dimensional parameters compared to the areas defined by canonical surfaces. In addition, a construction tree may not be available for the generation of this aerodynamic area. An illustration of the relationship between functional requirements and a blade shape is the performance optimization of this blade. Here, performance optimization can be reduced to shape optimization acting on the air flow around the

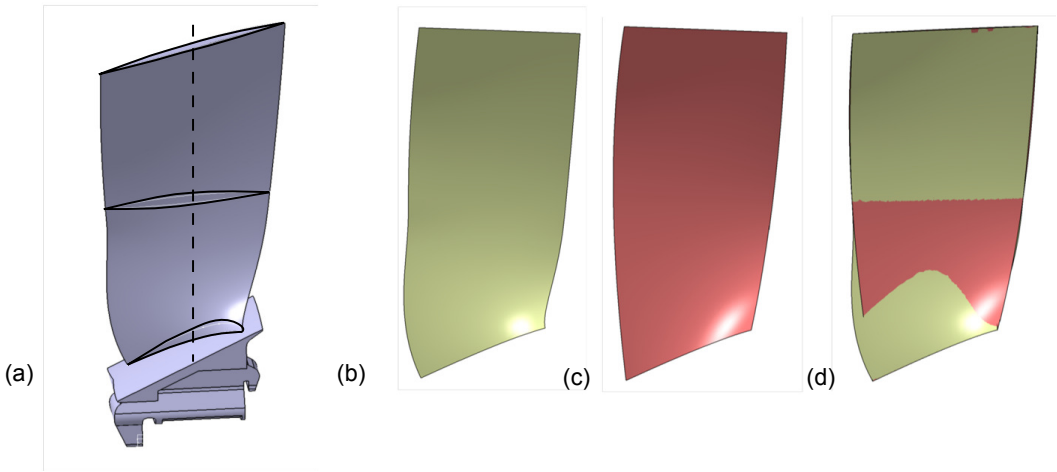


FIGURE 1.3: Examples of a fan blade subjected to functional parameters during a shape optimization process. Influence of shape generation algorithms across different software: (a) Initial fan blade model with the three sections used to optimize the blade shape, (b) aerodynamic area modeled with CAD software A, (c) aerodynamic area modeled with CAD software B and the same input parameters as software A, (d) superimposed aerodynamics showing their shape differences.

blade as well as on the mechanical behavior. Actually, the final shape is a compromise between the aerodynamic and mechanical requirements. Such an optimization process requires automated shape modifications, hence a parameterization with so-called design variables of the optimization problem. Consequently, the shape parameterization of the blade and the design variables input to the optimization process may differ from those used during the initial blade model generation process. The dimensions taken as design variables can be regarded as functional parameters. As a result, lacking a construction tree necessitates interactive modifications to set up the appropriate dimensions and the corresponding design variables. This configuration is somewhat similar to the configurations where construction trees are lost when geometric models are moved from one software environment to another one.

Parameterization for shape optimization incorporates the use of several models, often geometric models in a CAD environment and FE ones or finite volume ones in CAE environments; this is different from general purpose shape modifications. Such needs identify some geometric elements, i.e., faces and/or edges, of a model parameterization as reference data that monitor shape transformations and adaptations. As such, these geometric elements and their corresponding dimensions could be regarded as intrinsic to the functional requirements and the component they belong to, independent of any software, as long as similar modeling processes can be applied to generate this component from different software, i.e., CAD or CAE. However, this is subjected to the influence of geometric modeling algorithms and it can lead to fair differences between models generated from the same geometric elements and the same control parameters as shown in Figure 1.3. In Figure 1.3(d), the superimposed aerodynamic surfaces show a maximum distance of 3.2% of the total blade height between them. Because the same geometric data and control parameters have been fed in each software, it is internal algorithm parameters that influence the resulting surfaces. Notably, curve and surface parameterizations are among these internal parameters. Consequently, shape processing in a PDP that involves surface parameterization related to free-form surfaces is essentially valid within one software environment.

As pointed out, B-Rep NURBS models of free-form surfaces are not the only digital representation of a component during a PDP. Piecewise linear, i.e., faceted representations, are also frequently

used across a PDP for FE simulations, immersive simulations using virtual reality techniques for usage or maintenance scenarios, and so on. In addition to the use of a different type of representation, for each category of simulation the shape of the product component may differ because of the different simulation hypotheses and objectives. It is often the case for FE simulations where detail removal is a common task used to satisfy the process needs, thus creating different shapes of the same component. Consequently, increasing the diversity and/or amount of simulations derive even more different shapes and processing free-form surfaces across all these simulations and PDP stages increase significantly the amount of tedious tasks to transform them and meet the simulation requirements. Increasing the amount of interactive shape modification reduces the efficiency of a PDP with a significant cost [51]. Even though the previous considerations do not apply solely to free-form surfaces, the difficulties currently associated with free-form shape processing, i.e., creation without intuitiveness, tedious modifications, lack of robust shape exploration, significantly reduce the efficiency of PDPs. On a complementary basis, this analysis shows how advancements in free-form shape modeling and processing could improve some tasks and even more globally the efficiency of PDPs.

The variety of activities involved in the PDPs highlighted in Section 1.2.1 is tightly linked to the diversity of digital shape representations encountered throughout each of them. PDP processes are often centered on one of these multi-representations, because of its well suited aptitudes to describe a component for one PDP process, and consequently, a component model differs in shape between processes. For example, a CAD model of a component within the detailed design process is getting simplified and transformed into a mesh when adapted to an FE simulation. Performing the shape transformations to move a digital shape between two processes of a PDP is still an ongoing research topic awaiting significant progress. Underlying issues are the possible relationships between the geometric models and the possible mapping between geometric models when the shape of components differs between PDP processes.

Section 1.2.1 has highlighted the connection between geometric and semantic information. With regard to shapes, this connection refers to the concept of feature. Numerous works have addressed feature modeling and feature recognition for mechanical components. However, free-form surfaces embedded in these components have not significantly evolved and up to now the integration of such shapes is still difficult in industrial software. Similarly, setting up a tight connection between semantic information and free-form information is part of scientific issues. If robust connections between semantic information and simple geometric entities, i.e., planes, cylinders, have been proposed, they are developed slowly. Indeed, added semantic information is expanding but its tight connection with geometry is not yet benefiting shape modeling systems, i.e., functional modeling is still part of perspectives of the computer-aided tools. Additional issues and future trends are also discussed in Section 1.4.2.

1.3 Modeling and sketching free-form surfaces

The use of digital tools for the creation of 3-D models is now common practice. Different tools and approaches are available thanks to the decades of research and efforts in various sectors, including, in particular, computer-aided geometric design, computational geometry, computer graphics, signal and data acquisition and processing, and electronic engineering. Actually, while several years ago the only possibility to create a model was through computer-aided drafting or CAD systems, new methods now exist for the digital cloning of physical objects. They range from the well-established laser scanning acquisition methods to the recent use of digital photos for reconstructing digital representations of real objects. Each methodology exploits different geometric representations and

it better applies to specific users and applications. In the following subsections a brief overview of the main available methods, models, and functionalities for creating and modifying shapes is provided.

1.3.1 Computer-Aided Design systems for shape generation

The birth of computer-supported drafting and modeling systems dates from the beginning of the '60s in the last century. The first computer-aided system supporting the creation of digital models is the pioneering Sketchpad defined by Ivan Sutherland in his PhD thesis at MIT [52]. This system provided functions for 2-D interactive design using a light pen input device with a vector graphics display monitor. The interest in 3-D digital representation of objects emerged first in automotive and aerospace industries, for the production and the model reuse of such complex products made of a huge amount of recurrent components. Particularly, the ability to accurately manufacture free-form shapes like car body components was the focus and developed with the numerically controlled machine tools. De Casteljau at Citroen and Bézier at Renault defined almost in parallel methods for the definition and manipulation of 3-D curves and free-form surfaces which turned out to be equivalent and are still the basis of most of the current commercial CAD systems. The first commercial CAD systems were launched in the '70s and have had a larger development and distribution starting from the '80s thanks to the availability of UNIX workstations and later on to the capabilities of personal computers and laptops allowing the processing and rendering of the models. These systems have been mainly developed keeping in mind engineers as final users. Therefore, the capabilities for defining and changing the object shapes were more focused on the underlying geometric representation of the surfaces than on designers capabilities and intent. To try to get closer to specific users needs, according to a specific product domain, more specialized tools have been defined in the following years. In the free-form modeling field, we observed the creation of those referred as CAS or computer-aided industrial design (CAID) systems. They are specifically equipped with sophisticated functionalities for the creation, visual simulation, and rendering of 3-D surface-based objects. This includes, for instance, the rendering of the reflection of the lights, e.g., the so-called zebra strips (see Figure 1.4), and of the environment on the object for analyzing the quality of the surfaces and the effect that this object can evoke when inserted in the real world (see Figure 1.4). These systems were mainly thought for industrial stylists and were largely used in the automotive industry at the preliminary design phase. Despite their widespread usage, they were not directly used by stylists but by the so-called surfacers, i.e., technicians trained on these systems with the capability of correctly understanding the stylists drawings. Actually, designers still preferred the use of pen and paper for expressing their ideas and for achieving their final proposal. Indeed designers are mainly using scribbles to draw the initial ideas, which are then modified by successive refinements. Then, they correct the scribbles and add new elements for emphasizing dark and bright areas to provide shape details and appearance. This is quite a fuzzy and unpredictable process to browse and sharpen up potential solutions to their product ideas. Computer-aided systems weakly support such fuzzy processes that have required the insertion of precise information since the beginning. Thus, the common practices to create industrial design objects are either the creation of surfaces using specific section curves created by tracing the hand-made drawing of the product or by using reverse engineering techniques applied to a clay model created from the hand-made sketch [16] (see Figure 1.5).

Moreover, the models created with CAID systems very often required a conversion/refinement/re-creation in more engineering oriented systems to provide the correct digital representation suitable for the analysis activities at the embodiment design phase and the production activities at the process planning phase. Such re-generation in CAD systems was due to various reasons, both cultural and technical. Actually, creative users are mainly concentrating on the definition of beautiful shapes, not paying attention to the model specification, i.e., the underlying surface decomposition into patches, thus long and narrow patches can be generated during modeling operations, which may bring prob-

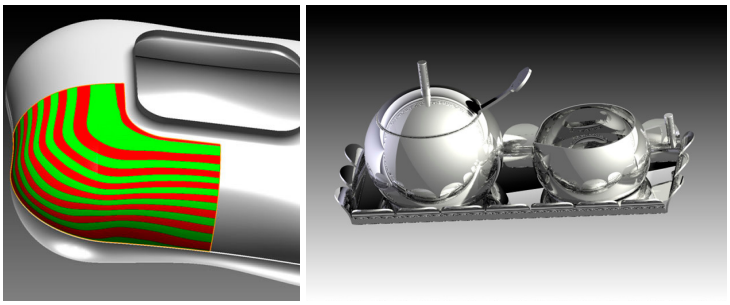


FIGURE 1.4: Example of zebra strips and of rendering (courtesy of Alessi).

FIGURE 1.5: Example of physical prototypes from hand-made sketches (courtesy of Eiger).

lems in later analysis and manufacturing phases. From the technical point of view, CAID systems did not provide engineering analysis and process planning tools, thus requiring a change of system with the related problems in the data transfer process. To overcome this limit, commercial systems providers introduced integrated environments in which the specialized modules can better communicate, see for instance, the CATIA portfolio [2].

Nowadays, the main efforts of free-form modeling systems are devoted to better support creative users at large, either in industrial manufacturing sectors or in the new entertainment industries, e.g., video games.

Preliminary solutions are emerging that support 3-D sketching for the creation of 3-D models directly from sketched elements. For instance, CATIA Natural Sketch [2] is re-applying Sutherland's approach allowing a designer to directly sketch with a pen on a monitor. If this technique is more advanced than traditional tools, it is still not immediate in directly achieving the 3-D model of any complex shape. These tools are mainly interpreting the sketched scribbles as potential construction curves of simple surfaces.

Other examples of domain-oriented 3-D systems for creative users are those developed for the edutainment industry. Initial 3-D animation dates from mid '80s with the very famous short movie *Luxo Jr.* [4]. Nowadays, 3-D models are widely used in movies and video games. In addition, with 3-D models being recognized as a good means for communicating information and for simulations, they have been getting a lot of attention in the media and communication sectors. Needs in these sectors are a bit different from those of the industrial engineering domain. Here, the important point is the visual appearance rather than a precise and detailed shape representation: textures are normally used to complete the visual details not always geometrically modeled. Additionally, animation can be another important action to be performed on the created models, thus requiring fast

shape transformations and adaptations. Due to these characteristics, these systems also allow the use of alternative discrete geometric representations in addition to parametric surfaces, i.e., NURBS or B-Splines as adopted in engineering-oriented systems and described in the following subsection.

Despite all these efforts, mainly devoted to providing more user-oriented simplified modeling functionalities, these systems are still mainly using traditional modeling operations, such as lofting, sweeping, curve interpolation, filleting, and blending. At the same time, they are still missing, or at best weakly supporting, the association of semantic information with the elements of the modeled objects. Such an association is crucial for efficient modifications, consistency maintenance, interoperability within and between PDP processes, automation of simulations and process planning, as well as model re-use and retrieval.

1.3.2 Digital Shape generation from real objects

An alternative way to create digital objects processes the data acquired from real objects. This approach gained widespread usage in various disciplines ranging from medicine to archeology, from entertainment to site planning [13, 27, 15]. Even engineering tasks adopted it as reverse engineering technology. A typical use is the creation of a digital representation of a product when this representation is not available, either because the CAD model no longer represents the real object, e.g., changes occurred on the physical object, or it does not exist at all. This typically happens to support maintenance tasks of structures and other equipments. Reverse engineering takes place also at any stage of a design process where a physical prototype is created either for analyzing an initial idea or for tangibly evaluating a validated solution before its production. It is also used to assess competitors' products for a better understanding of their components, behavior, and cost assessments.

The category of technology adopted for real data acquisition strongly depends on the type of objects to be reconstructed and on the planned use of their digital model. The physical objects can be measured using coordinate-measuring machines (CMM), laser scanners, structured light digitizers, or industrial computed tomography (CT) scanners. For instance, for medical purposes as well as for the reconstruction of underground equipment, computed tomography is adopted, which allows the capture of 3-D images of internal elements by using X-rays.

More recently, low-cost solutions emerged using technologies from the game industry, e.g., Kinect, or plain photographs from portable cameras, e.g., 123D Catch available on iPad [1] and processed by the Autodesk cloud or 3DSOM Pro [3]. Generally, these solutions are providing rough models in terms of geometry visually enriched with textures corresponding to the acquired pictures. Thus, they are currently not usable for engineering applications. Other approaches include the use of calibrated cameras , which employ photogrammetry methods for the reconstruction of shapes. These methods use digitized points combined with 2-D images for the recovery of additional information [34].

The reconstruction of mechanical objects has been traditionally obtained through the use of 3-D modeling technologies. The measured data, usually represented as a points cloud plus additional images, need to be further processed to obtain the watertight 3-D CAD representation useful for design and production purposes. Normally, the overall process involves the use of intermediate object representations in terms of discretized triangle meshes obtained after a sequence of semi-automatic steps: registration of the acquired data; reduction of the number of points; mesh generation and repair [10, 56, 34]. The generation of a CAD model from the resulting mesh is a semi-automatic process too. Generally, it is a trial-and-error process to obtain the best fit with CAD patches. In some cases, i.e., when the surface does not incorporate sharp features for instance, this process can become automatic [21].

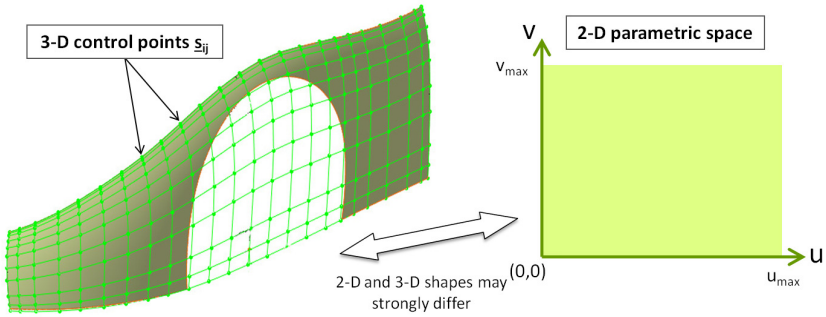


FIGURE 1.6: Example of a trimmed B-Spline surface with its control point network and associated parametric space wherein the trimming lines are defined.

1.3.3 Object representations

As described in the previous sections, depending on the specific purposes of a PDP phase, various simulation analyses and tools are applied, each of them working on specific object descriptions. Therefore, different representation schemes are used across a PDP. As already mentioned, the first representation scheme for free-form surfaces, named after its developer Bézier, has been defined during the '60s in the automotive industry. It exploits matrix calculus, thus providing the advantage of handling transformations, such as rotation or translation through simple and efficient operations on matrices. Thus, surfaces can be expressed as a combination of polynomials, also called blending functions, which specify the contribution of the defining points of their control polyhedrons. Points on the surfaces can be evaluated at specific values of the two parameters specified in their rectangular parametric domain. Bézier surfaces have a strong limit for complex objects definition: no local modification capability is available; that is to say, the transformation of any control point has an effect on the overall surface, which can be mitigated only by using a very large number of control points. This is unfeasible because the degree of the surface is directly linked to the number of control points. To overcome this serious drawback, object boundary decomposition into patches is mandatory. To weaken this constraint, B-splines and non-rational uniform B-splines (NURBS) have been introduced [22, 41] – they are the de-facto standard in current CAD/CAS systems. They provide a polynomial piecewise definition of a surface, which automatically maintains certain continuity conditions among the polynomial pieces. In addition, differently from Bézier and B-Splines, NURBS are also able to represent functional surfaces (planes, cylinders, spheres, and so on) thus guaranteeing the use of a single representation for all the possible shapes. In all these schemes, the modification of a surface can be indirectly obtained by moving its control points (see Figure 1.6). This can be non-trivial and requires a fair understanding of the influence of control points movements over the surface modifications. To this end, more intuitive methods for direct shape modifications have been developed and are still under investigation as discussed in the next section of this chapter. Moreover, the definition of complex shaped objects cannot be easily performed using solely parametric quadrangular surface, i.e., patches. Thus, the description of complicated shapes necessitates also trimmed patches, i.e., only a part of the parametric domain of a patch is effectively used to define the object boundary. For example, the detailed specification of a car body can require more than 6,000 patches (see Figure 1.7, where a subset of 80 patches is displayed). To represent high quality and globally smooth shapes, these patches have to be smoothly connected, thus adding continuity constraints which should be preserved during all the shape modifications . These continuity constraints increase the complexity of shape modification processes.

Another possibility to represent surfaces uses implicit surfaces. An implicit surface consists of those points in the 3-D space that have the same potential. Thus, it is defined by a function

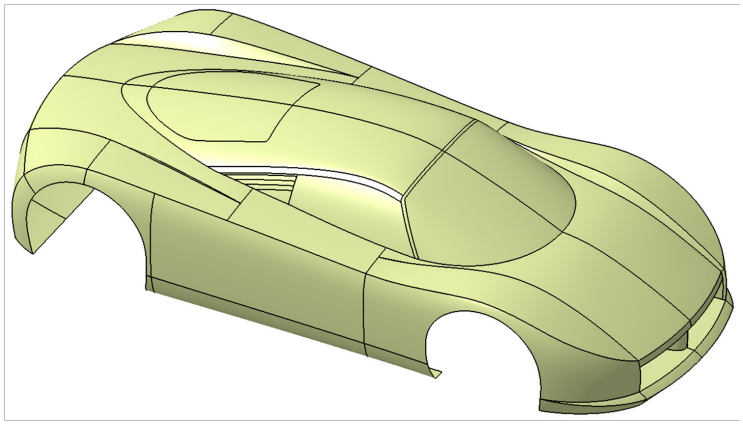


FIGURE 1.7: Example of trimmed B-spline surfaces used to create a car body (courtesy of Fiores II [23]).

$f(x,y,z) = 0$ linking the three point coordinates. The function f can contain any mathematical expression. When polynomial, it gives rise to the so-called algebraic surfaces, which have gained interest in the CAD domain to overcome problems related to specific types of computation, such as intersection, and to data exchange [20, 9]. Nevertheless, they have not been widely used in CAD systems because of the difficulty of both converting traditional representations into algebraic forms of a reasonable degree and defining suitable means for controlling their shape.

As an alternative to these continuous representations, tessellated models are used in several PDP phases, such as virtual reality simulations, rapid prototyping, FE analyses. They are also the first model created when building the digital cloning of physical objects. They provide an explicit, even if approximated, representation normally in terms of triangles. In this way, even if not unique, tessellated models overcome the problem of decomposing the object boundary into suitable patches. They also gained a lot of attention in recent years for modeling purposes due to the increased storage and computational efficiency of computers in handling large data sets. Thus, analysis and modeling capabilities traditionally defined for continuous models are being defined for discrete models as well [11]. The major drawback of such models is related to the lack of intuitive shape control parameters, which makes it hard to control a shape during their definition and editing.

Subdivision surfaces are another kind of representation recently introduced in CAD systems. They can be considered in between continuous and discrete representations. Introduced in the late '70, subdivision surfaces have recently gained major attention. They can be considered as a generalization of spline surfaces, since they are also controlled by a coarse control mesh but, in contrast to spline surfaces, they enable the representation of arbitrary topology-sided surface patches. Subdivision surfaces are generated by repeated refinements of control meshes: after each topological refinement step, the positions of the (old and new) vertices are adjusted based on a set of local averaging rules. Figure 1.8 shows an example of different levels of iterative refinements of the subdivision surface control mesh [11].

Subdivision surfaces offer the advantage of representing an object with a single surface, thus there is no need to manage additional topological and continuity conditions as in the case of splines. In contrast, they cannot represent surfaces for which the triangulation is not the result of repeated refinement of a coarse control mesh. These arbitrary surfaces have to be remeshed to subdivision connectivity in a preprocessing step. Such a remeshing corresponds to a resampling of the surface, and it usually leads to sampling artifacts and loss of information. Artifacts are unexpected features on the subdivision surface which cannot be intuitively controlled by repositioning the vertices of the original coarse mesh. Additionally, they are missing a pipeline of a constructive approach as

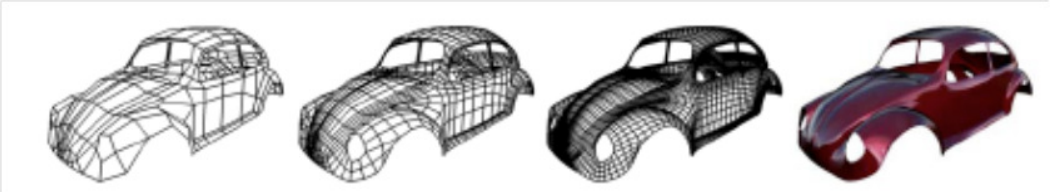
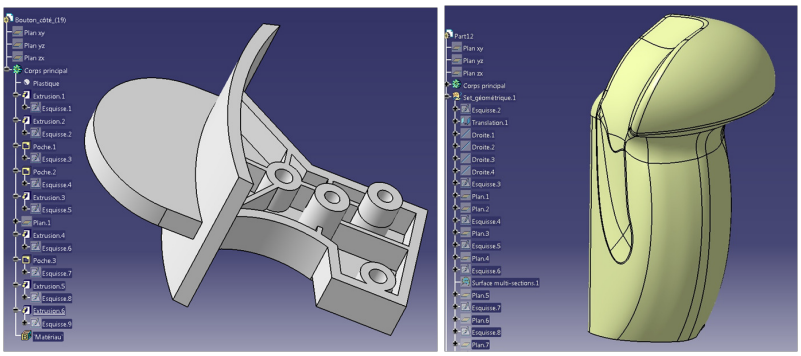


FIGURE 1.8: Subdivision surface generated by iterative refinement of the original control mesh [11].



(a) (b)

FIGURE 1.9: Examples of object modeling using: (a) modeling operations and a construction tree as available in current CAD modelers, (b) general free-form surfaces where parameterization and construction tree concepts are not available, i.e., some modeling operations cannot be edited and the modeler that collects operations exists but there is no chronology reflecting a construction process.

available for NURBS models. This is mainly due to their insufficient shape predictability and, thus, to the difficulty in their control network choice. Moreover, the insertion of G^1 discontinuities, i.e., sharp edges, or functional surfaces is difficult to handle. Therefore, subdivision surfaces better apply when the main modeling objective is the creation of smoothly appearing objects or when shapes have to be transmitted with a low band, as for animation, than in the cases where a strong control on the surface accuracy and quality is needed.

The so-called T-splines are an alternative to subdivision surfaces in creating non-rectangular surface patches and, at the same time, they enable local modifications without the insertion of additional control points. T-splines are now being inserted in some commercial tools like Rhino [45, 44]. Anyhow, several issues are still to be tracked to get the same operational maturity as traditional representations. Other representations, not discussed here, are adopted in other processes during a PDP, such as tetrahedral meshes for FE simulations and voxel decompositions for topological optimization. This diversity of representations generates difficulties when adapting and linking them to the various processes' needs of a PDP. Associating semantic information to a product shape may facilitate such an integration and adaptation. Additionally, creating tools that rely on the same metaphor applied to different representations could definitively speed up some processes of a PDP.

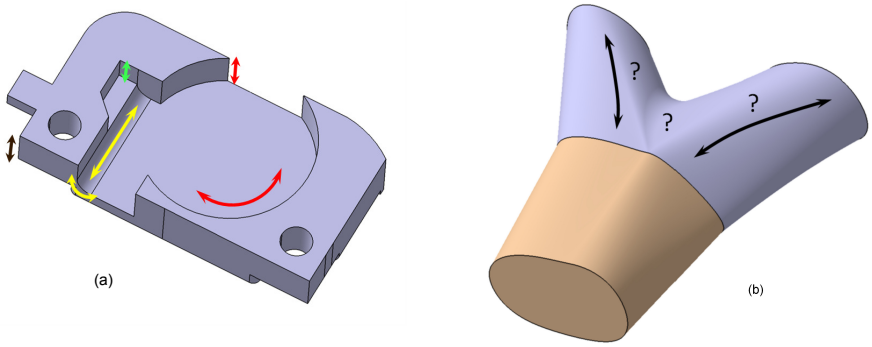


FIGURE 1.10: (a) a mechanical component with some modeling operations highlighted as gesture-based operations with arrows. Arrows illustrate possible translational or rotational movements defining locally a primitive feature possibly used to produce the shape locally, and (b) a example of free-form object where the gray area indicates a configuration with no single gesture capable of describing this entire area.

1.3.4 Shape modification

Shapes can be created from scratch or, more often, through modifications of existing models since most products correspond to refined versions of existing ones. As seen in Section 1.2, the final solution of a product is normally achieved through successive modifications, including shape transformations. Thus, the importance of efficient shape modification tools is straightforward.

In mechanical engineering, to ease the edition of shapes, parametric modeling as well as the concept of feature have been introduced and widely adopted for shape design. Features correspond to recurrent shape primitives that can be instantiated with few intuitive parameters. These parameters have a predictive behavior when their values change during a PDP process. As already mentioned, they provide a way to associate some application semantics, e.g., manufacturing or assembly operations, surface finishing [47]. Examples of mechanical features are extrusions, holes, pockets, fillets, ribs, and so on. To satisfy application requirements, i.e., a process in the PDP devoted to a specific product, current CAD systems provide users with means for the specification of feature classes in addition to those already available. This approach is further enriched with the possibility to directly edit object elements, i.e., faces, edges, and vertices, to stretch their shape, through the so-called direct modeling paradigm that does not require any construction tree.

The mechanical engineering field provides designers with construction tree structures in CAD and CAE software to ease shape modifications and associate a process to a digital representation of an object (see Figure 1.9(b)). Industrial CAD software rely on an incremental B-Rep (Boundary representation) modeling paradigm where volume modeling is performed iteratively using planar sketched contours subjected to either extrusion or revolution operations, as generative parameters, and to either material addition or removal, as shape sculpting parameters. The construction tree structure is based on reference planes containing the sketched contours and primitive shapes defined from the generative and shape sculpting parameters. Constructive solid geometry (CSG) fits also in this approach where regularized Boolean operators act as shape sculpting parameters. Parameterized contours and parameterized extrusion lengths or rotation angles are helpful to propagate shape modifications throughout a construction tree to modify an object shape (see Figure 1.9(a)).

Indeed, these modeling operations all can be aggregated as gesture-based interactions and, more precisely, as single trajectory-based operations (see Figure 1.10(a)). Effectively, extrusions relate to simple translational movements, revolutions to rotational ones. Even more complex modeling func-

tions like blending radii, which can produce complex surfaces, may be reduced to single trajectory operations when decomposed into a connected sequence of edges forming a chain or a single loop. The sequence of edges forms the trajectory that can be assigned to these blending operations. Extrusions with draft surfaces or blending with a variable radius are extensions of the previous functions where the initial contour used in the primitive definition is assigned an evolution law, often a simple one. However, the concept of single trajectory is still applicable and the concept of gesture that follows this trajectory, even if it is not the effective user interaction, is also the common concept that aggregates all these modeling operations.

When it comes to more general free-form surfaces that are not covered by the previous categories, the concept of gesture-based modeling operations still exists but is not so straightforwardly applied. Figure 1.10(b) illustrates a configuration where the gray area of the object cannot be associated with a single hand gesture that would cover the description of this entire area. Indeed, this area must be decomposed into several gestures. Consequently, there no modeling function that can generate the gray area in one step. The use of bifurcating trajectories does not provide a better shape insight either. If not the only explanation, the observations about gestures illustrate the extent to which it becomes more difficult to define a construction tree and its corresponding modeling process when considering free-form surfaces. Similarly, the diversity of input parameters used by the previous operators has not led to a generic structure of modeling process. This is also due to the intrinsic level of degree freedom in the specification of free-form shapes which cannot be characterized by few parameters and rules. Consequently, free-form shape modifications are generally tedious because operator parameterization and construction tree are totally or partly missing. Consequently, free-form shapes are still defined as a set of free-form surfaces resulting from low-level modeling and modification operations. Approaches for the manipulation of free-form surfaces can be classified according to the type of operations and associate control parameters. Thus, two main approaches can be distinguished depending on whether they use:

- operators that do not apply a deformation to an initial shape/surface. This category can be further refined depending on whether they use a set of:
  - numerical and/or geometric parameters, e.g., curves and points, to control classical procedures like sweep, loft, extrusion, intersection, trimming, projection, blending, filleting, and so on. These operators can be used for the definition of the overall structure of the objects' shapes as well as for local refinements;
  - points to find the surfaces that best fit them with respect to specific criteria such as the least-squares minimization. This is the method applied in RE to create digital models.
- operators deforming an initial shape/surface [26, 42]. This category can be further refined depending on whether the deformation is considered as a:
  - global deformation controlled by global parameters. These methods are widely used in animation where there is no need to have a dimensionally accurate shaping of the objects;
  - local and global deformations through the use of some geometric parameters and constraints, like curves and geometric continuity, to be preserved within and between the deformed and unchanged areas of the object. The free-form feature approach we developed belongs to this category and is further described in Section 1.4.1.

The simulation driven approach belongs to this second category, i.e., the geometric deformation of a shape is associated with a mechanical behavior much in the same way an FE simulation is coupled with shape transformations in a shape optimization process (see the example in Section 1.2.2). This approach provides suggestions on possible shapes by proposing alternatives which are somehow optimal with respect to specific boundary conditions, i.e., prescribed forces and displacements,

required to specify the mechanical behavior associated with the geometric deformation process. This approach is being inserted into commercial tools, e.g., solidThinking [5].

This categorization could be further refined depending on the type of underlying surface representation: NURBS surfaces, subdivision surfaces, meshes, voxels, and so on. For the sake of conciseness, we mainly focus on free-form surfaces by means of NURBS curves and surfaces since this type of representation is the core definition of CAD models. Furthermore, subcategories could also be defined depending on whether they can also use data and parameters extracted from images [34]. Even if this is a promising way to reconstruct shapes from real objects, these approaches will not be detailed here. Finally, the control parameters can refer to classical ones, i.e., dimensions, geometric constraints (coincidence and so on), or to higher-level parameters linking aesthetic as well as emotional properties to the underlying geometric models through a set of constraints and dedicated objective functions. These aspects are further developed in the next section.

Despite the categories of operators described above, fine shapes adjustments can still require trials and errors because there is no satisfactory result achieved with the set of high-level operators available in a given CAD software. This means acting on non-intuitive surface parameters related to the way a surface is defined mathematically, e.g., B-spline control points. This is even more complex when changes are performed on models created by others: the model decomposition into surface patches is not unique, it depends on the designers modeling technique during the model creation, which may not reflect the shape structure. As a consequence, designers can hardly express modifications based on the desired shape changes. Indeed, designers need to deal either with a limited set of rather basic operators or they must directly address the low-level parameters, e.g., control points, of the shape model being modified. Therefore, users often need to add new control points or surface patches or they have to perform trimming operations to be able to localize the shape changes or to add specific functional surfaces and/or sharp edges. These modifications have the effect of creating more and more complex models, which makes the propagation of modifications a critical issue when processing objects incorporating free-form surfaces. In addition, continuity conditions across patches are complementary constraints of free-form surface modifications that add further complexity to this process. Also, it has to be noticed that possible unpredictable effects of geometric operators may happen, which is another source of complexity to monitor free-form shape modification processes. Finally, additional difficulties are faced by engineers when geometric operators fail, e.g., no solution to a blend generation, self-intersections appear when generating a surface through interpolating sections. This detailed analysis about free-form shape generation and modification processes enforces some open issues raised in Section 1.2.2.

In conclusion, despite significant improvements in free-form design tools, modifications are still too dependent upon the underlying geometric model of an object rather than addressing effectively its shape. Therefore, modifications are generally time-consuming and require a lot of expertise to predict the effects of some control parameter changes over the object shape. Thus, new methods have to be developed to allow a user to interact with a shape through more intuitive and meaningful concepts, independent of the underlying patch decomposition and geometric representation of an object.

1.4 Toward advanced models, methods, and tools for modeling free-form surfaces

The previous sections highlighted several issues and problems of the free-form surface modeling approaches that have been developed during the last decades. From this analysis, it is clear that the so-called feature-based approach can play a key role for the modeling of shapes within a PDP

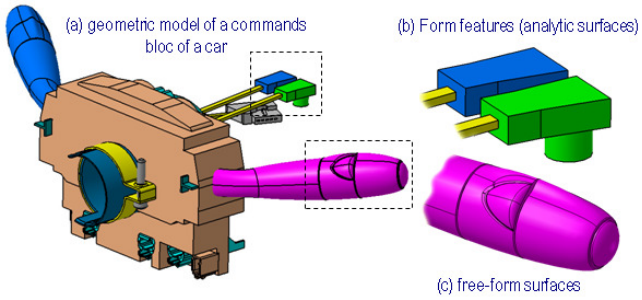


FIGURE 1.11: (a) Geometric model of a commands bloc of a car designed with (b) form features (analytic surfaces) and (c) free-form surfaces (courtesy of PSA).

framework. Therefore, this section reviews the recent advances that have been developed to extend the feature concept to free-form shapes, thus giving rise to several free-form features modeling strategies. More precisely, Section 1.4.1 reviews the state-of-the-art of those free-form feature-based approaches and Section 1.4.2 presents a clear and detailed analysis of the open issues and future trends.

1.4.1 Free-Form Features - A promising way to the modeling of free-form surfaces within a PDP

The free-form shape design operators, as available in CAD/CAS software, are not yet intuitive enough to model objects incorporating free-form shapes and they do not enable fast and easy shape modifications. To overcome these limits, some researches have been undertaken to try to adapt the feature concept to the modification of objects containing free-form shapes.

In the context of a feature-based approach, the geometric model of an object is not anymore perceived as a collection of vertices, edges, and faces but, as a well-organized set of features corresponding to slots, ribs, stiffeners, and so on. In the mechanical engineering domain, features have been implemented as shape constructive primitives classified in accordance with their shared characteristics. However, form features based on free-form surfaces have not been extensively studied. Figure 1.11(b) shows examples of form features. All the properties related to a feature type are specified within a feature class that defines a template for all its instances. This always includes the generic shape of the feature, and a number of parameters, e.g., length, width, and constraints, e.g., parallel, perpendicular, that describe this shape. By specifying values for the parameters, an instance of the feature class can be created and then added to a feature model [14]. A feature-based model usually contains the description of the part in terms of its constituting feature instances, the information related to their mutual dependencies and the chronology of construction. Thus, a user does not manipulate directly the geometric entities but the features themselves through a set of control parameters that can be modified later on. The use of features is also crucial when considering all the information that has to be associated to the parts all along the PDP. In fact, features can be associated to some application-specific meaning or processing. Examples are the process planning activity in which features can provide useful information for the choice of most suitable machining operations and tools. Similarly, shape adaption for FE simulation purposes can be sped up by removing meaningless features. Features addressed in this case are based on shape characteristics, e.g., holes, and mechanical parameters, e.g., finite element size, boundary conditions. To provide a product description in terms of the various PDP activities, feature recognition strategies have been developed [46, 31, 28], and are mandatory when evolving from one process to another in a PDP because each process or task defines a specific context and often requires specific features. To carry

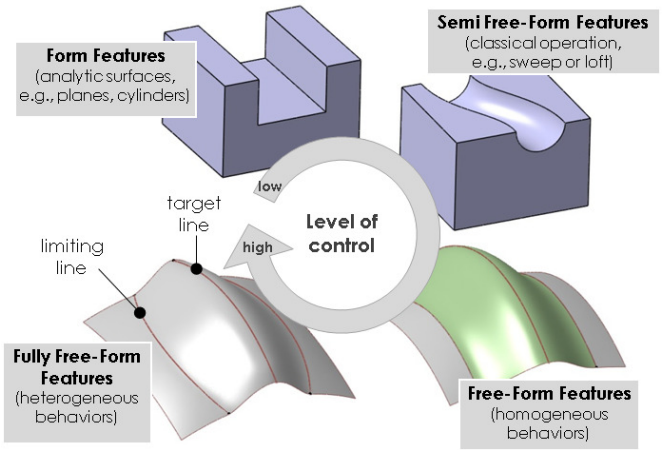


FIGURE 1.12: Features classification based on the concept of level of control [37].

out appropriate analyses and comparisons between the existing feature-based approaches, suitable evaluation criteria and an appropriate ranking system have to be adopted. The criteria define the framework for the analyses. They reflect the application domain needs and characteristics. Considering the aesthetic and engineering design of products, four main categories of criteria have been identified [37]:

- criteria related to the adopted geometric models and feature types, which characterize:
  - the nature of the manipulated representations (parametric, polyhedral, and so on);
  - the dependencies on the underlying geometric models to identify shape-oriented approaches that can be applied to different underlying representations;
  - the type of topology preservation, i.e., whether the underlying surface decomposition into patches is modified or not;
  - the type of discontinuities that can be handled (prescribed or free if they can be induced by a shape modification process).
- criteria related to shapes and functions that distinguish approaches according to:
  - the use of shape-oriented feature classification;
  - the parameterization capabilities by means of curves or numerical values;
  - the number of control parameters as well as the way validity constraints can be checked.
- criteria related to the capacity to tune the changes:
  - handling local and/or global shape modifications;
  - the level of freedom in the shape definition;
  - the level of control of the DOF including the possibility to set heterogeneous behaviors.
- criteria addressing the interactivity to characterize approaches whose manipulations are tedious as well as approaches which do not incorporate high-level operators.

Based on these evaluation criteria, three main categories of features involving free-form surfaces can be distinguished depending on the level of control they have on the resulting shapes (see Figure 1.12):

- semi free-form features defined by free-form surfaces obtained by classical rules, such as loft/sweep operations, interpolation rules or specific relationships directly expressed between the control points. The user has few possibilities to control the resulting shapes. This category can be further refined depending on the type of constraints used to define the features:
  - point-based approaches [8, 50, 57, 58] define semi free-form features from a set of key points which often directly correspond to the control points of the underlying Bézier patches;
  - curve-based approaches [33, 53, 54] define parameterized free-form surfaces from a set of parameterized free-form curves;
  - finally, surface-based approaches [18, 17, 55] define semi free-form features directly from a bivariate function $z = f(x,y)$ which significantly reduces the range of possible applications.
- free-form features defined by free-form surfaces, obtained through the use of adequate techniques expressing a homogeneous behavior over the whole surface [23]. Freedom is greater but handling the modified area is not always as free as possibly required by users. The insertion of features inside the initial model often requires topological changes, e.g., trimming operations, which is always risky [24, 25].
- fully free-form features characterized by a higher level of freedom in the definition of the shapes obtained through the use of techniques prescribing heterogeneous behaviors over different areas of free-form surfaces. The modified area can be bounded by any free-form curve while preserving the surface initial topology (no trimming operations or patches insertion). Such features take advantage of the NURBS description power [41] and fit well stylists' requirements.

More precisely, a fully free-form deformation feature (δ-F^4) corresponds to an area of a free-form surface that results from a deformation under specific constraint lines: the target, the limiting, and the driving lines [35]. The use of curves as main feature parameters is reflecting the way stylists normally design their shapes either for providing the overall object structure or for inserting the various details [24]. The target lines give the global directions of a shape, whereas the limiting lines are free-form curves used to monitor this shape around the target lines and bound the deformation area (see Figure 1.13(a)). To avoid changes in the surface topology while localizing the deformation, an automatic fixation algorithm is used to block the control points that have an influence outside the deformation area. Driving lines correspond to curves used to drive the repetition of more simple shapes (see Figure 1.13(b)). A restricted set of high-level parameters are used to control both the shape of the curves and the shape of the surface around these curves. They include numerical parameters, e.g., position, dimensions and orientation, shape control parameters (type of multi-minimizations), and the type of discontinuity (G^0, G^1). The numerical parameters are used to parameterize the features, the multiple minimizations enable the specification of heterogeneous shape behaviors [39] and the discontinuities can be inserted along any free-form directions expressed through constraint lines. Additionally, shape constraints, such as internal planar areas within a specific feature, can be included. The various parameters used to control the constraint lines can be instantiated in two ways by using the concept of feature template: either designers directly specify their own constraint lines or they can instantiate numerical parameters (dimensions, angles, shape behaviors, and so on) used to shape the constraint lines with the generic deformation engine. Once defined over the surface, δ-F^4 can be manipulated in two ways: either in a prescriptive manner, while modifying the different parameters that have been instantiated (dimensions, shape behavior, and so on), or in a qualitative manner, while using a set of interactive deformation operators, e.g., push, pull. To select the δ-F^4 to be instantiated and the associated template in [38], a complete feature classification has been proposed.

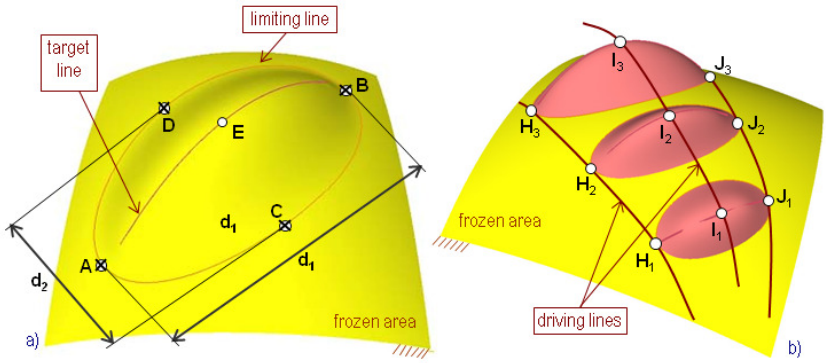


FIGURE 1.13: A basic shape feature: (a) defined by target and limiting lines parameterized with few numerical parameters and heterogeneous deformation behaviors and (b) A complex shape feature defined as a repetition of BSF along several driving lines.

The strength of this approach lies in the perfect harmony between degree of flexibility and shape control. The concept of δ-F^4 template enables mastered manipulations of the free-form curves that are prescribed as constraints for the free-form surface deformation process. This double level of deformation (curve deformation followed by surface deformation) enables an easy access to a wide variety of shapes without limiting designers' creativity. Moreover, the use of multi-minimizations introduces a new level of shape control, thus justifying even more the term of fully free-form deformation features in comparison with the other approaches dealing with semi free-form features. The main drawback of this approach lies in the fact that solely some basic validity conditions can be defined and managed through a rudimentary validation process.

Additionally, the deformation engine used and most of the feature parameters can be applied to any kind of models [37], thus making the approach particularly suited for various processes in a PDP. Anyhow, even if such an approach is already a good way to help designers and stylists specifying shapes, there is still a gap between the available tools and the way people think. This has been widely studied within the AIM@SHAPE European Network of Excellence [6] from which a structured decomposition of shape information has been proposed (see Figure 1.14). Actually, shapes can be decomposed into three distinct levels which interact with each others: the geometric level defining the mathematical model of the shape, the structure level which helps decompose the shape into areas on which specific information can be associated, and the third level which gathers together the semantic information. Therefore, and as described in Figure 1.14, there is a need for developing top-down approaches that would manipulate shapes by means of semantics. To this aim, several methods have been proposed [29, 30] to try to set up a new approach where designers and stylists manipulate curves through a set of higher level control parameters like the straightness, the convexity or the acceleration of the curves [23]. Also, new approaches that would take into account the emotional aspects are foreseen.

In conclusion, adopted right from the beginning within an integrated semantic-based approach, the use of δ-F^4 is very promising. Not only does it help the designers and stylists interacting with shapes without focusing on the underlying mathematical models, but the feature concept is also a good way to express the semantics used all along the PDP. However, there are still open issues and future trends that are developed in the next section.

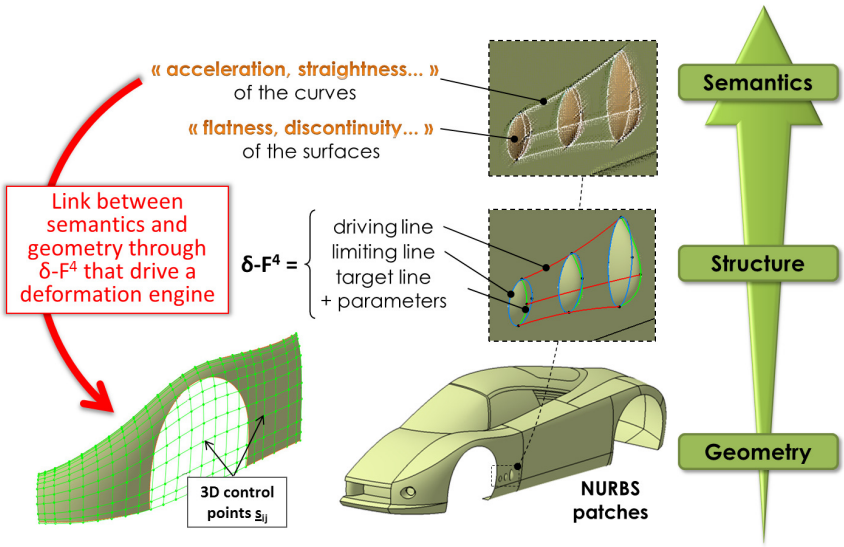


FIGURE 1.14: Structured decomposition of shapes and link between semantics and geometry through the use of δ-F^4 that drive a deformation engine.

1.4.2 Open issues and future trends

Even if the above-mentioned approaches have deeply improved the way engineers can interact with free-form surfaces within a PDP framework, there are still many open issues as well as future trends.

Clearly, if some real improvements have been proposed in this domain, it is still difficult to take into account the uncertainties which are intrinsic to the activities of engineers and designers when defining their shapes. For example, the uncertainty the designer has when defining the blending areas between the different constraint lines may generate undesired undulations. To provide a user-friendly tool, which does not force the designer to be very precise, the possibility to relax automatically the end points of the target lines has been offered [40] as well as the possibility to relax the areas surrounding the limiting lines [35]. However, a deeper analysis of the possible configurations generating undesired surface behaviors has to be undertaken. Actually, the final shapes result from the integration of various constraints defined step by step by the different stakeholders of a PDP. However, the global optimization problem is often not completely formalized and the final shapes often result from tedious manipulations and exchanges between the involved stakeholders. The resulting systems of constraints are still very hard to analyze, i.e., they can be over- or under-constrained either globally or locally. Thus, new methods and tools have to be developed to help the designers understand and fix over-constrained configurations [32] involving geometric models as well as technological parameters.

Moreover, even if the feature concept is a real improvement to designer's interactions with shapes, its use is often confined to the modeling of solids and its extension to the multi-representations of a product is not yet fully supported. Though construction trees are efficient to parameterize and edit a shape, they suffer from non-uniqueness and inability to be transferred among successive phases of a PDP. Evolving from a construction tree to a construction graph containing all non-trivial construction trees of an object is a means to describe an intrinsic structure of an object. This approach can be combined with the ability to automate the generation of this graph from a given

object shape. As a result, it provides a powerful tool to analyze and modify an object's shape, and it is actively contributing to the object structure [6]. Presently, this is being developed and applied to engineering components bounded by simple surfaces [12]. In the free-form shape domain, the integration of structural and detail features is not fully supported and it is still difficult to associate a construction tree to define a construction graph for these shapes. However, this open issue is being addressed with more simple engineering components to evaluate the complexity of construction graph generation. Future trends will head for higher complexity to process shapes incorporating free-form surfaces.

Many of the current limits can be ascribed to the underlying well-known NURBS mathematical model whose capabilities are not fully exploited. The weights, the knot sequences as well as the trimming lines are so many parameters often considered as fixed during the modification steps. For example, most of the deformation techniques implemented in the current CAD software do not act on the trimming lines, which remain static in the parametric spaces of the patches. Thus, as it has already been tested for a single patch, considering the control points of the 2-D trimming lines as unknowns of the deformation process could help solve over-constrained configurations [36]. Moreover, there exist many issues regarding the parameterization of those free-form surfaces as well as their decomposition into patches. Still, the available modification mechanisms are often time-consuming and far from the users' needs. Of course, this slows down the progress of a PDP where all its processes should be run as quickly as possible to fit the end-user requirements. Such a new way for designing shapes requires the definition of a completely new approach much more oriented toward semantics and emotions rather than the low-level constraints and the underlying mathematical models. At the end, shapes should really be seen as a result of an emerging process integrating the numerous constraints manipulated by the different stakeholders of a PDP. In this sense, a first attempt toward the definition of a semantic-based approach has been proposed for the deformation of free-form curves [29]. In this work, the deformation results from the solution of an optimization problem where the constraints are directly acting on meaningful aesthetic quantities, e.g., straightness, acceleration of the curves, are close to the way designers and stylists think. This semantic-based modeling approach is now being extended to the definition and manipulation of free-form surfaces. In the future, the link to the emotional impact is also foreseen so that the surfaces may be directly shaped to answer the end-user emotional requirements.

Along with these developments, new challenges have to be taken into account that arise from the use of new interesting low-cost virtual reality devices putting the end user at the center of some PDP simulation processes. Other issues that complement the present list have already been mentioned in Section 1.2.

References

[1] http://www.123dapp.com.

[2] http://www.3ds.com/products/catia/welcome/.

[3] http://www.3dsom.com/.

[4] http://www.pixar.com/short_films/theatrical-shorts/luxo-jr.

[5] http://www.solidthinking.com/.

[6] Aim@Shape. http://www.aimatshape.net. *Advanced and innovative models and tools for the development of semantic-based systems for handling, acquiring, and processing knowledge*

embedded in multi-dimensional digital objects. European Network of Excellence, Key Action: 2.3.1.7 Semantic-based Knowledge Systems, VI Framework, 2004.

[7] T. Amabile. *Creativity in Context: Update to the Social Psychology of Creativity*, West View Press, Boulder, CO, 1996.

[8] C.K. Au and M.M.F. Yuen. "A Semantic Feature Language for Sculptured Object Modelling", *CAD*, 32(1):63–74, 2000.

[9] C. Bajaj, J. Bloomenthal, B. Guo, J. Hart, and G. Wyvill. "Implicit Surfaces for Geometric Modelling and Computer Graphics", In *SIGGRAPH'96 Course Notes*, 1996.

[10] P. Benk, R.R. Martin, and T. Varady. "Algorithms for Reverse Engineering Boundary Representation Models", *CAD*, 11:839–851, 2001.

[11] M. Botsch, M. Pauly, C. Rossl, S. Bischoff, and L. Kobbelt. Geometric Modeling Based on Triangle Meshes, In *ACM SIGGRAPH 2006 Courses*, 2006.

[12] F. Boussuge, J-C. Léon, S. Hahmann, and L. Fine. "Extraction of Generative Processes from B-Rep Shapes and Application to Idealization Transformations", *CAD*, 46:79–89, 2014.

[13] C. Bradley and B. Currie. "Advances in the Field of Reverse Engineering", *Computer-Aided Design and Applications*, 2(5):697–706, 2005.

[14] W.F. Bronsvoort, R. Bidarra, and P. Nyirenda. "Developments in Feature Modelling", *Computer-Aided Design and Applications*, 3(5):655–664, 2006.

[15] M. Callieri, M. Dellepiane, P. Cignoni, and R. Scopigno. *Processing Sampled 3-D Data: Reconstruction and Visualization Technologies*, CRC Press, Boca Raton, FL, Taylor and Francis, 2011.

[16] C.E. Catalano, B. Falcidieno, F. Giannini, and M. Monti. "A Survey of Computer-Aided Modeling Tools for Aesthetic Design", *ASME Journal of Computing and Information Science in Engineering*, 2:11–20, 2002.

[17] J.C. Cavendish. "Integrating Feature-Based Surface Design with Free-Form Deformation", *CAD*, 27(9):703–711, 1995.

[18] J.C. Cavendish and S.P. Marin. *A Procedural Feature-Based Approach for Designing Functional Surfaces*, H. Hagen, editor, SIAM, Philadelphia, PE, 1996.

[19] N. Cross. *Engineering Design Methods*, John Willey, Chichester, UK, 3rd Ed., 2000.

[20] T. Dokken and B. Juttler. *Computational Methods for Algebraic Spline Surfaces: An Exploratory Workshop*, Springer, Berlin, Heidelberg, New-York, 2005.

[21] M. Eck and H. Hoppe. "Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type", In *ACM SIGGRAPH 1996*, 325–334, 1996.

[22] G. Farin, J. Hoschek, and M.-S. Kim. *Handbook of Computer Aided Geometric Design*, Elsevier Science B. V. Amsterdam, 2002.

[23] Fiores. http://www.fiores.com. *Formalization and integration of an optimized reverse engineering styling workflow. Brite-Euram Project N. BE96-3576, 1997-99*, 1997.

[24] M. Fontana, F. Giannini, and F. Meirana. A Free-Form Features Taxonomy, In *Proceedings of Eurographics'99*, volume 3, 107–118, 1999.

[25] M. Fontana, F. Giannini, and F. Meirana. "Free-Form Features for Aesthetic Design", *International Journal of Shape Modeling*, 6(2):273–302, 2000.

[26] J. Gain and D. Bechmann. "A Survey of Spatial Deformation from a User-Centered Perspective", *ACM Transactions on Graphics*, 27(107):1–21, 2008.

[27] X.W. Gao, Y. Qian, and R. Hui. "The State of the Art of Medical Imaging Technology: from Creation to Archive and back", *Open Medical Informatics Journal*, 5(1):73–85, 2011.

[28] F. Giannini. "Geometric Reasoning as a Tool for Integrating Design and Production Activities in Mechanical Environment, *International Journal of Shape Modeling*, 2(2-3):117–137, 1996.

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

[30] F. Giannini, M. Monti, J. Pelletier, and J-P. Pernot. "A Survey to Evaluate how Non Designers Perceive Aesthetic Properties of Styling Features", *Computer-Aided Design and Applications*, 10(1):129–138, 2012.

[31] J.H. Han, M. Pratt, and W.C. Regli. Manufacturing Feature Recognition from Solid Models: A Status Report, *IEEE Trans. On Robotics and Automation*, 16(6):782–796, 2000.

[32] A. Lee Saint John and J. Sidman. "Combinatorics and the Rigidity of CAD Systems", *CAD*, 45:473–482, 2013.

[33] P.J. Nyirenda, M. Mulbagal, and W.F. Bronsvoort. "Definition of Free-Form Surface Feature Class", *Computer-Aided Design and Applications*, 3(5):665–674, 2006.

[34] M. Panchetti and J-P. Pernot andP. Véron. "Towards Recovery of Complex Shapes in Meshes using Digital Images for Reverse Engineering Applications", *CAD*, 42(8):693–707, 2010.

[35] J-P. Pernot, B. Falcidieno, F. Giannini, and J-C. Léon. "Fully Free-Form Deformation Features for Aesthetic Shape Design", *Journal of Engineering Design*, 16(2):115–133, 2005.

[36] J-P. Pernot, B. Falcidieno, F. Giannini, and J-C. Léon. A Hybrid Models Deformation Tool for Free-Form Shapes Manipulation, In *Proceedings of the 34th Design Automation Conference (ASME DETC08)*, 2008.

[37] J-P. Pernot, B. Falcidieno, F. Giannini, and J-C. Léon. "Incorporating Free-Form Features in Aesthetic and Engineering Product Design: State-of-the-Art Report", *Computers in Industry*, 59(6):626–637, 2008.

[38] J-P. Pernot, F. Giannini, B. Falcidieno, and J-C. Léon. Parameterized Free-Form Feature Templates, In *Proceedings of the IEEE Shape Modelling International Conference (SMI'09)*, 2009.

[39] J-P. Pernot, S. Guillet, J-C. Léon, B. Falcidieno, and F. Giannini. "Shape Tuning in Fully Free-Form Deformation Features", *ASME Journal of Computing and Information Science in Engineering*, 5(1):95–103, 2005.

[40] J-P. Pernot, Q. Qian, and P. Véron. *Constraints Automatic Relaxation to Design Products with Fully Free-Form Features*, Springer Verlag, 2007.

[41] L. Piegl and W. Tiller. *The NURBS Book*, Springer Verlag, 1997.

[42] R. Raffin. *Free-Form Deformations or Deformations Non-Constrained by Geometries or Topologies*, Hidalgo et al., editors, Springer Verlag, 2013.

[43] R Schmidt, A Khan, G Kurtenbach, and K Singh. On Expert Performance in 3-D Curve-Drawing Tasks, In C. Grimm and J. J. LaViola Jr., editors, *EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling*, 2009.

[44] T.W. Sederberg, G.T. Finnigan, X. Li, and H. Lin. "Watertight Trimmed NURBS", *ACM Transactions on Graphics*, 27(3), 2008.

[45] T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. "T-Splines and T-NURCCS", *ACM Transactions on Graphics*, 22(3):477–484, 2003.

[46] J J. Shah, D. Anderson, Y. S Kim, and S. Joshi. A Discourse on Geometric Feature Recognition from CAD Models, *ASME Journal of Computing and Information Science in Engineering*, 1:41–51, 2001.

[47] J.J. Shah and M. Mäntylä. *Parametric and Feature-Based CAD/CAM*, Wiley-Interscience Publication, John Wiley Sons, Inc., 1995.

[48] A. Shahwan, J-C. Léon, G. Foucault, M. Trlin, and O. Palombi. "Qualitative Behavioral Reasoning from Components' Interfaces to Components' Functions for DMU Adaption to FE Analyses", *CAD*, 45(2):383–394, 2013.

[49] C. Shao, A. Bousseau, A. Sheffer, and K. Singh. "CrossShade: Shading Concept Sketches Using Cross-Section Curves", *ACM Transactions on Graphics, SIGGRAPH*, 2012.

[50] Y. Song, J.S.M. Vergeest, and I. Horvàth. Reconstruction of Free-Form Surface with Parameterized Features, In *ASME DETC'02*, 2002.

[51] M. Stiteler. Chaps: Construction History and Parametrics, Technical report, Advance Technology Institute, January 2004.

[52] I.E. Sutherland. *"Sketchpad: A Man-Machine Graphical Communication System"*, PhD thesis, MIT, Boston, MA, 1963.

[53] E. van den Berg and W.F. Bronsvoort. "Parameterized, Constraint-Based Wrapping of Free-Form Shapes", *Computers & Graphics*, 31(1):89–99, 2007.

[54] E. van den Berg, H.A. van der Meiden, and W.F. Bronsvoort. Specification of Free-Form Features, In *ACM Symposium on Solid Modeling and Applications*, 56–64, 2003.

[55] P.A. van Elsas and J.S.M. Vergeest. "Displacement Feature Modelling for Conceptual Design", *CAD*, 30(1):19–27, 1998.

[56] T. Varady, M.A. Facello, and Z. Terek. "Displacement Feature Modelling for Conceptual Design", *CAD*, 39(5):379–388, 2007.

[57] J.S.M. Vergeest and I. Horvàth. Reconstruction of Free-Form Surface with Parameterized Features, In *Proceedings of the IEEE Shape Modeling International Conference*, 20–28, 2001.

[58] G. Vosniakos. "Investigation of Feature-Based Shape Modelling for Mechanical Parts with Free-Form Surfaces", *International Journal of Advanced Manufacturing Technology*, 15(3):188–199, 1999.